



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO - UFES
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Explorando a diversidade de pares de caminhos disjuntos por arestas em redes ópticas elásticas sobreviventes

Marina Girolimetto

Orientador: Prof. Dr. Marcia Helena Moreira Paiva
Universidade Federal do Espírito Santo - UFES

Coorientador: Prof. Dr. Claunir Pavan
Universidade Federal da Fronteira Sul - UFFS

VITÓRIA-ES
ABRIL/2018

Marina Girolimetto

**Explorando a diversidade de pares de caminhos
disjuntos por arestas em redes ópticas elásticas
sobreviventes**

Explorando a diversidade de pares de caminhos disjuntos
por arestas em redes ópticas elásticas sobreviventes

Linha de pesquisa: Telecomunicações e Tecnologia da
Informação (TTI)

Orientador: Prof. Dr. Marcia Helena Moreira Paiva
Universidade Federal do Espírito Santo -
UFES

Coorientador: Prof. Dr. Claunir Pavan
Universidade Federal da Fronteira Sul -
UFFS

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO - UFES
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
VITÓRIA-ES
ABRIL/2018

Marina Girolimetto

Explorando a diversidade de pares de caminhos disjuntos por arestas em redes ópticas elásticas sobreviventes

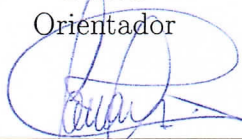
Explorando a diversidade de pares de caminhos mais
curtos disjuntos por arestas em redes ópticas elásticas
sobreviventes

Trabalho aprovado. Vitória-ES, 26 de Abril de 2018

M. Paiva

Prof. Dr. Marcia Helena Moreira

Paiva
Orientador



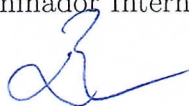
Prof. Dr. Claunir Pavan

Coorientador

Maria Claudia Silva Boeres

Prof. Dr. Maria Claudia Silva Boeres

Universidade Federal do Espírito Santo
Examinador Interno



Prof. Dr. Leandro Colombi Resendo

Instituto Federal do Espírito Santo
Examinador Externo

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO - UFES
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
VITÓRIA-ES
ABRIL/2018

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Setorial Tecnológica,
Universidade Federal do Espírito Santo, ES, Brasil)

G526e Girolimetto, Marina, 1992-
Explorando a diversidade de pares de caminhos disjuntos
por arestas em redes ópticas elásticas sobreviventes / Marina
Girolimetto. – 2018.
83 f. : il.

Orientador: Marcia Helena Moreira Paiva.
Coorientador: Claunir Pavan.
Dissertação (Mestrado em Engenharia Elétrica) –
Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Algoritmos. 2. Redes ópticas elásticas. 3. Sobrevivência
de Rede. 4. Proteção de caminho dedicado. 5. Caminhos
disjuntos. I. Paiva, Marcia Helena Moreira. II. Pavan, Claunir.
III. Universidade Federal do Espírito Santo. Centro Tecnológico.
IV. Título.

CDU: 621.3

Agradecimentos

Gostaria de agradecer à minha mãe que me deu forças em todos os momentos e superou a distância que nos separava.

Ao meu pai que acreditou em mim.

À minha orientadora Prof Dr. Marcia Helena Moreira Paiva que sempre deu seu máximo para me orientar. Agradeço pela paciência e ensinamentos.

Ao Prof Dr. Marcelo Eduardo Vieira Segatto que não mediu esforços para me trazer para a Universidade Federal do Espírito Santo, minha eterna gratidão.

Ao meu co-orientador Prof Dr. Claunir Pavan que me orienta desde a graduação. Agradeço imensamente todo o apoio e ensinamentos todos esses anos.

Ao meu colega do LabTel Dr. Rodrigo Stange Tessinari por toda ajuda e suporte em diversos momentos do meu mestrado.

Ao Prof Dr. Fábio de Oliveira Lima pela ideia do trabalho e auxílio para o desenvolvimento do mesmo.

À todos os meus colegas e amigos do LabTel que me ajudaram e apoiaram.

À toda minha família que me deu suporte para os estudos.

À todos meus amigos que sempre estavam disponíveis para me ouvir e apoiar nos momentos difíceis.

À CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), à FAPES (Fundação de Amparo à Pesquisa e Inovação do Espírito Santo) e o projeto FUTEBOL (*Federated Union of Telecommunications Research Facilities for an EU-Brazil Open Laboratory*) pela bolsa de estudos concedida em determinados momentos ao decorrer do mestrado.

“Como um jardineiro que prepara o canteiro, à espera da semente oportuna, em meu coração resiliente, faz brotar a mais bela flor de maio, perdurando por todas as estações da minha vida.” (Anderson Fernando)

Resumo

Em redes ópticas elásticas, a alta taxa de transmissão de bits pode causar um grande problema de perda de dados quando ocorre falha de enlace ou nodo na rede. Para manter a rede disponível e protegida de falhas, diferentes esquemas de proteção podem ser aplicados. No esquema de proteção de caminho dedicado, são definidos caminhos de trabalho e de *backup*, que não compartilham recursos, para cada par de nodos através de algoritmos que encontram caminhos disjuntos. Geralmente, o algoritmo de *Suurballe e Tarjan* é utilizado para buscar pares de caminhos disjuntos por arestas tais que a soma dos comprimentos dos caminhos seja mínima. No entanto, para um mesmo par de nodos origem e destino, podem existir diversos pares de caminhos de mesmo comprimento mínimo, mas com diferentes comprimentos de caminho de trabalho e de *backup*.

Neste trabalho, duas versões do algoritmo de Suurballe e Tarjan são propostas para lidar com essa diversidade. Para cada par de nodos origem e destino de uma dada topologia de rede, essas versões encontram o menor par de caminhos de trabalho e de *backup* mais balanceado, com caminhos com comprimentos com a menor diferença possível, e menos balanceado, com caminhos com comprimentos com a maior diferença possível. Ambos algoritmos são testados e analisados em um conjunto de 40 topologias *2-aresta-conexas* de redes de telecomunicações reais. O impacto destes algoritmos é investigado através do coeficiente de proteção e do número de *transponders* necessários para esquemas de proteção de caminho dedicado. Os resultados mostram uma diferença de até 29% nos comprimentos dos caminhos de trabalho entre as estratégias nas redes testadas. Além disso, foi avaliado o comportamento dos algoritmos em cenários de tráfego dinâmico e proteção de caminho dedicado, sob simulações considerando falha única de enlace. Com isso, foi identificado a partir das redes estudadas, que a taxa de bloqueio (métrica de desempenho) de cada algoritmo, dependerá da topologia física da rede.

Palavras-chave: Redes Ópticas Elásticas. Sobrevivência de Rede. Proteção de Caminho Dedicado. Menores Caminhos Disjuntos por Arestas. Algoritmos de Roteamento.

Abstract

In elastic optical networks, high bit rate can cause a major data loss problem when a network link or node failure occurs. To keep the network available and protected from failures, different protection schemes can be applied. In the dedicated path protection scheme, working and backup paths are defined, which do not share resources, for each pair of nodes through algorithms that find disjoint paths. Generally, Suurballe and Tarjan's algorithm is used to find shortest pairs of edge-disjoint paths such that the sum of the path lengths is minimal. However, for the same pair of source and destination nodes, there may be diverse shortest pairs of paths of the same minimum length, but with different working and backup paths lengths.

In this work, two versions of the Suurballe and Tarjan's algorithm are proposed to deal with that diversity. For each pair of source and destination nodes of a given network topology, these versions find the most balanced shortest pair of working and backup paths, with paths lengths with the smallest possible difference, and the least balanced, with paths lengths with the greatest possible difference. Both algorithms are tested and analyzed in a set of 40 2-edge-connected topologies of real-world optical telecommunication networks. The impact of these algorithms is investigated through the protection coefficient and the number of transponders required by dedicated-path protection schemes. Results show a difference of up to 29% in the working paths lengths between the strategies in the networks tested. Also, we evaluate the behavior of the algorithms in scenarios of dynamic traffic and dedicated path protection scheme, under simulations considering single link failures. Thus, it was identified from the studied networks that the blocking rate (performance metrics) of each algorithm will depend on the physical topology of the network.

Keywords: Elastic Optical Networks. Network Survivability. Dedicated Path Protection. Edge-disjoint Shortest Paths. Routing Algorithms.

Lista de Figuras

Figura 1 – (a) Grafo conexo. (b) Grafo desconexo.	2
Figura 2 – Uma rede <i>2-conexa</i> para a qual, se o menor caminho entre os nodos O e D (o caminho de três saltos em linha tracejada vermelha) é escolhido como caminho de trabalho, não existe caminho disjunto para ser usado quando uma falha ocorrer.	4
Figura 3 – Dois pares de caminhos disjuntos por arestas com comprimento mínimo igual a $C = 10$ interconectando o nodo origem O e o nodo destino D . Em (a), um caminho de trabalho com 3 saltos (linha pontilhada vermelha) e um caminho de <i>backup</i> com 7 saltos (linha cheia azul). Em (b), um caminho de trabalho e um outro de <i>backup</i> com 5 saltos cada.	5
Figura 4 – Três caminhos ópticos separados por bandas de guarda. Adaptado de [Christodouloupoulos, Tomkos e Varvarigos 2011].	10
Figura 5 – Arquitetura do enlace e do nodo numa rede óptica elástica, apresentando o <i>BVT</i> e o <i>BV – WXC</i>	11
Figura 6 – Exemplo referente às restrições de contiguidade e continuidade do <i>RSA</i> . <i>Slots</i> disponíveis estão representados pela cor branca e <i>slots</i> ocupados estão representados pela cor cinza.	14
Figura 7 – Mecanismos de sobrevivência em redes [Saini e Garg 2013].	16
Figura 8 – Arquitetura da ferramenta <i>ElasticO++</i> . Adaptado de [Tessinari et al. 2016].	21
Figura 9 – Módulos de controle da ferramenta <i>ElasticO++</i> . Adaptado de [Tessinari et al. 2016].	22
Figura 10 – Exemplo gráfico de uma simulação de uma topologia no <i>software ElasticO++</i>	25
Figura 11 – (a) Topologia de rede; De $v_o = v_1$ para $v_d = v_8$: (b) menor caminho (em linha tracejada azul) obtido com o algoritmo de <i>Dijkstra</i> , (c) atualização dos pesos das arestas de acordo com o passo da “transformação de custos”, (d) segundo menor caminho (em linha com ponto e traço rosa), obtidos com o algoritmo de <i>Dijkstra</i> , (e) caminhos disjuntos de trabalho e de <i>backup</i> obtidos com o algoritmo <i>ST</i>	27
Figura 12 – Processo de busca de K menores caminhos do par de nodos $v_o = v_1, v_d = v_5$. Adaptado de [Martins e Pascoal 2003].	29
Figura 13 – Fluxograma do algoritmo Suurballe e Tarjan Mais Balanceado (<i>STMB</i>).	32
Figura 14 – Topologia <i>2-aresta-conexa</i> exemplo.	34

Figura 15 – Dois caminhos disjuntos por arestas com comprimentos $R = 8$ (em linha tracejada rosa) e $S = 8$ (em linha com ponto e traço azul) do par origem v_8 para o destino v_{17}	34
Figura 16 – Dois caminhos disjuntos por arestas com comprimentos $R = 6$ (em linha tracejada rosa) e $S = 10$ (em linha com ponto e traço azul) do par origem v_8 para o destino v_{17}	35
Figura 17 – Dois caminhos disjuntos por arestas com comprimentos $R = 3$ (em linha tracejada rosa) e $S = 13$ (em linha com ponto e traço azul) do par origem v_8 para o destino v_{17}	36
Figura 18 – Dois caminhos disjuntos por arestas com comprimentos $R = 5$ (em linha tracejada rosa) e $S = 11$ (em linha com ponto e traço azul) do par origem v_8 para o destino v_{17}	37
Figura 19 – Intervalo entre falhas e duração das falhas. Adaptado de [Wang, Zeng e Zhao 2003].	42
Figura 20 – (a) Rede Arnes, com 17 nodos e 20 enlacs; (b) Rede Cernet, com 29 nodos e 45 enlacs.	46
Figura 21 – Diferença normalizada dos caminhos de trabalho do <i>STMB</i> e do <i>STLB</i> para todas as redes. As redes estão organizadas pelo número de nodos em ordem crescente.	47
Figura 22 – Caminhos de trabalho e de <i>backup</i> definidos pelos algoritmos <i>STMB</i> e <i>STLB</i> na rede “Arnes” para os pares de nodos origem o e destino d	47
Figura 23 – Diferença normalizada entre os comprimentos dos caminhos de trabalho e de <i>backup</i> do <i>STMB</i> ($\Theta_{o,d}^+$) e <i>STLB</i> ($\Theta_{o,d}^-$) da rede “Arnes”. O eixo horizontal representa pares de origem e destino para os quais existe diferença nos comprimentos de caminhos entre as duas estratégias. O par apresentado na Figura 22 está circulado. Os pares estão organizados de acordo com $\Theta_{o,d}^-$ em ordem crescente.	48
Figura 24 – Diferença normalizada entre a soma dos comprimentos dos caminhos de trabalho e a soma dos comprimentos dos caminhos de <i>backup</i> no <i>STMB</i> (γ^+) e <i>STLB</i> (γ^-). O eixo horizontal representa as 40 redes analisadas. As redes estão organizadas pelo seu número de nodos em ordem crescente.	49
Figura 25 – Diferença normalizada entre o número de <i>transponders</i> usados nos caminhos de trabalho do <i>STMB</i> e <i>STLB</i> para todas as redes. As redes estão organizadas pelo número de nodos em ordem crescente.	50
Figura 26 – Coeficiente de proteção (k_p) para cada rede usando os algoritmos <i>ST</i> , <i>STMB</i> e <i>STLB</i>	51
Figura 29 – Taxas de bloqueios de contiguidade e continuidade do <i>ST</i> , <i>STMB</i> e <i>STLB</i>	55

Figura 30 – Taxas de bloqueios do <i>ST</i> , <i>STMB</i> e <i>STLB</i> referentes a $\lambda = 200$, $T_i = 1, 0$, $T_h = 0, 6$, e 10^5 requisições.	56
Figura 31 – Taxas de bloqueios do <i>ST</i> , <i>STMB</i> e <i>STLB</i> referentes a $\lambda = 100$, $T_i = 1, 0$, $T_h = 0, 6$, e 10^6 requisições.	57
Figura 27 – Taxa de bloqueio vs. taxa de chegada das redes Arnes (a) e Cernet (b), para $T_i = 1$ e $T_h = 0, 05s$. Taxa de bloqueio vs. intervalo entre falhas das redes Arnes (c) e Cernet (d), para $\lambda = 100$ e $T_h = 0, 05s$. Taxa de bloqueio vs. duração da falha das redes Arnes (e) e Cernet (f), para $\lambda = 100$ e $T_i = 1s$	59
Figura 28 – Considerando a maior diferença de taxa de bloqueio do <i>ST</i> com relação à do <i>STMB</i> e do <i>STLB</i> : (a) Coincidência de caminhos entre o <i>ST</i> , o <i>STMB</i> e o <i>STLB</i> ; (b) Coincidência de comprimentos de caminhos entre o <i>ST</i> , o <i>STMB</i> e o <i>STLB</i> ; Considerando a menor diferença de taxa de bloqueio do <i>ST</i> com relação à do <i>STMB</i> e do <i>STLB</i> : (c) Coincidência de caminhos entre o <i>ST</i> , o <i>STMB</i> e o <i>STLB</i> ; (d) Coincidência de comprimentos de caminhos entre o <i>ST</i> , o <i>STMB</i> e o <i>STLB</i>	60

Lista de Tabelas

Tabela 1	–	Número de <i>slots</i> de frequência de 12,5 Ghz para a modulação DP-QPSK [Tessinari, Colle e Garcia 2017] e taxa de transmissão de bit. . .	23
Tabela 2	–	Conjunto de topologias de redes reais [Pinto 2017] e algumas de suas características: número de nodos e de enlaces, grau nodal médio, tempo de processamento em segundos no <i>STMB</i> e no <i>STLB</i> . As topologias estão ordenadas pelo número de nodos.	45
Tabela 3	–	Taxas de bloqueio no <i>ST</i> , no <i>STMB</i> e no <i>STLB</i> e taxas de bloqueio de tipos de bloqueio: de contiguidade, de continuidade e de falta de recursos.	55
Tabela 4	–	Taxas de bloqueio no <i>ST</i> , no <i>STMB</i> e no <i>STLB</i> com $\lambda = 200$, $T_i = 1, 0$, $T_h = 0, 6$, e 10^5 requisições.	56
Tabela 5	–	Taxas de bloqueio no <i>ST</i> , no <i>STMB</i> e no <i>STLB</i> com $\lambda = 100$, $T_i = 1, 0$ e $T_h = 0, 6$, e 10^6 requisições.	56

Lista de Abreviaturas e Siglas

- BPSK *Binary Phase Shift Keying* (Deslocamento de Fase Binária)
- BV-WXC *Bandwidth Variable Wavelength Cross-Connects* (Comutador Óptico de Largura de Banda Flexível)
- BVSSS *Bandwidth Variable Spectrum Selective Switches* (Comutadores Seletivos de Espectro de Largura de Banda Flexível)
- BVT *Bandwidth Variable Transceivers* (Transponder de Largura de Banda Flexível)
- CERNET2 *China Education and Research NETwork2* (China Educação e Pesquisa Rede2)
- CSV *Comma Separated Values* (Valor Delimitado por Vírgula)
- DP-QPSK *Dual-Polarization Quadrature Phase Shift Keying* (Deslocamento de Fase em Quadratura e Dupla Polarização)
- EON *Elastic Optical Network* (Redes Ópticas Elásticas)
- ITU *International Telecommunication Union* (União Internacional de Telecomunicações)
- OEO *Optical-Electrical-Optical* (Óptico-Elétrico-Óptico)
- OFDM *Orthogonal Frequency-Division Multiplexing* (Multiplexação por Divisão de Frequência Ortogonal)
- QAM *Quadrature Amplitude Modulation* (Modulação de Amplitude em Quadratura)
- QPSK *Quadrature Phase Shift Keying* (Deslocamento de Fase em Quadratura)
- RMSA *Routing, Modulation and Spectrum Assignment* (Roteamento, Modulação e Atribuição de Espectro)
- ROADM *Reconfigurable Optical Add/Drop Multiplexer* (Multiplexadores Ópticos *Add-Drop* Reconfiguráveis)
- RSA *Routing and Spectrum Assignment* (Roteamento e Atribuição de Espectro)
- RWA *Routing and Wavelength Assignment* (Roteamento e Alocação de Comprimento de Onda)
- SBVT *Sliceable Bandwidth Variable Transceivers* (*Sliceable* Transponder de Largura de Banda Flexível)

ST *Suurballe and Tarjan* (Suurballe e Tarjan)

STLB *Suurballe and Tarjan Least Balanced* (Suurballe e Tarjan Menos Balanceado)

STMB *Suurballe and Tarjan Most Balanced* (Suurballe e Tarjan Mais Balanceado)

WDM *Wavelength-Division Multiplex* (Multiplexação por Divisão de Comprimento de Onda)

Sumário

1 – Introdução	1
1.1 Conceitos de Grafos	2
1.2 Proteção em Redes Ópticas Elásticas	3
1.3 Objetivo Geral	7
1.4 Objetivos Específicos	7
1.5 Contribuições	7
1.6 Estrutura da Dissertação	8
2 – Redes Ópticas Elásticas	9
2.1 Introdução	9
2.2 Arquitetura das Redes Ópticas Elásticas	10
2.2.1 Arquitetura de Nós	12
2.3 Roteamento e Atribuição de Espectro	13
2.4 Fragmentação	15
2.5 Mecanismos de Sobrevivência	15
2.5.1 Esquemas de Proteção	16
2.5.1.1 Proteção de Enlace e de Caminho	16
2.5.1.2 Proteção Dedicada	16
2.5.1.3 Proteção Compartilhada	17
2.6 Transponders Ópticos e o Coeficiente de Proteção na Proteção de Caminho Dedicado	18
2.7 Conclusão	19
3 – Ferramentas e Algoritmos	20
3.1 ElasticO++	20
3.1.1 Arquitetura do ElasticO++	21
3.1.2 Processo de Requisição e Alocação de Recursos	23
3.1.3 Contribuições na Ferramenta <i>ElasticO++</i>	24
3.2 Algoritmo de Suurballe e Tarjan	25
3.3 Algoritmo de Yen	27
3.4 Conclusão	30
4 – Algoritmos Propostos	31
4.1 Suurballe e Tarjan Mais Balanceado	31
4.2 Suurballe e Tarjan Menos Balanceado	35
4.3 Tempo Computacional	37

4.4	Vantagens e Desvantagens	38
4.5	Conclusão	38
5	Metodologia dos Testes	39
5.1	Introdução	39
5.2	Procedimentos Metodológicos	39
5.2.1	Testes: Etapa 01	40
5.2.1.1	Métricas Para Identificar Diferenças Entre Comprimentos do STMB e STLB	40
5.2.2	Testes: Etapa 02	41
5.2.3	Testes: Etapa 03	41
5.3	Conclusão	43
6	Resultados	44
6.1	Cenário de Testes	44
6.2	Comportamento e Diferenças do STMB e do STLB	46
6.2.1	Diferença Normalizada do STMB e STLB nas 40 Redes de Teleco- municações Reais	46
6.2.2	Diferenças do STMB e STLB Relativamente a C da Rede Arnes	47
6.2.3	Diferenças do STMB e STLB Relativamente a C das 40 Redes de Telecomunicações Reais	49
6.3	Impacto do STMB e STLB no Número de Transponders e no Coeficiente de Proteção	50
6.4	Simulação de Falhas: Utilização de Recursos versus Carga de Tráfego	51
6.5	Simulação de Falhas: Utilização de Recursos versus Intervalo Entre Falhas	52
6.6	Simulação de Falhas: Utilização de Recursos versus Duração da Falha	52
6.7	Tempo de Simulação da Rede USA no STMB e no STLB	52
6.8	Taxa de Bloqueio do ST na Rede Cernet	53
6.9	Conclusão	57
7	Conclusão e Trabalhos Futuros	61
	Referências	63

1 Introdução

Nos últimos anos, o tráfego na *Internet* vem obtendo um crescimento contínuo, principalmente após o surgimento de aplicações de alta velocidade como serviços multimídia, *TV* de alta definição, computação em nuvem, entre outros. As redes ópticas de transporte de telecomunicações tradicionais baseadas em Multiplexação por Divisão de Comprimento de Onda (*WDM*) estabelecem conexões com canais espaçados a uma distância fixa de $50GHz$ ou $100GHz$ [Chatterjee, Sarma e Oki 2015]. Contudo, a flexibilidade dessas redes é restrita e quando a demanda de tráfego ocupa menos que o tamanho fixo ocorre desperdício de largura de banda. Portanto, uma evolução na tecnologia no campo das redes de telecomunicações precisou ser realizada. Assim surgem as redes ópticas elásticas (*EON*), através da busca por soluções de uma rede que ajuste dinamicamente os seus recursos tais como, a largura de banda óptica, espectro e o formato de modulação (que permite o ajuste de largura de banda) de acordo com os requisitos de cada conexão [Tomkos et al. 2014].

As redes ópticas elásticas fornecem um serviço de largura de banda que varia de acordo com a demanda de tráfego, acomodando apenas a largura de banda suficiente para o tráfego; garantem o crescimento de velocidade através da agregação de largura de banda, que combina várias portas físicas em um comutador/roteador dentro de uma única porta lógica; e suportam a expansão e contração da largura de banda dinâmica [Chatterjee, Sarma e Oki 2015].

Para alocação de recursos de espectro em uma rede *EON*, é necessário resolver o problema do Roteamento e Atribuição de Espectro (*RSA*), sendo o espectro óptico, o espaço disponível em um enlace dividido em partes chamadas de *slots*. No *RSA*, uma conexão estabelece um caminho entre um par de nodos origem e destino para encaminhar o tráfego necessário. Este problema deve buscar em uma conexão: uma rota através de um algoritmo de roteamento e *slots* livres na rota definida para a alocação de requisições de conexão. Para isso, duas restrições devem ser cumpridas: contiguidade e continuidade de espectro. Para satisfazer a restrição de contiguidade, os *slots* alocados devem estar posicionados um ao lado do outro; já na restrição de continuidade, a posição definida para alocar o *slot* no primeiro enlace de um caminho óptico, deve ser a mesma para todos os outros enlaces. Este problema também pode ser resolvido juntamente com a modulação, sendo chamado de Roteamento, Modulação e Atribuição de Espectro (*RMSA*) [Christodouloupoulos, Tomkos e Varvarigos 2011]. O *RMSA* adiciona a atribuição do formato de modulação ao espectro no enlace, enquanto o *RSA* não lida com esta parte. Neste trabalho, o *RSA* foi empregado através da ferramenta *ElasticO++* [Tessinari et al. 2016].

O *ElasticO++* é um *framework* que permite testes em roteamento, modulação,

atribuição de espectro, desfragmentação, parâmetros físicos da rede como atenuação, potência de entrada e de saída, entre outros e topologias, proporcionando também o desenvolvimento de novos algoritmos e técnicas para redes ópticas elásticas. Ele será abordado melhor no Capítulo 3 [Tessinari et al. 2016].

1.1 Conceitos de Grafos

As topologias de redes ópticas elásticas de telecomunicações podem ser modeladas por grafos, onde os vértices representam os nodos, e as arestas representam os enlaces entre os nodos. A seguir, são apresentadas as definições de algumas medidas de grafos consideradas neste trabalho [Boaventura-Netto 2006, Whitney 1932]:

- A ordem de um grafo é o número de vértices do grafo;
- O grau de um vértice é a quantidade de arestas que incidem no vértice e o grau nodal médio considera a média dos graus de todos os vértices do grafo;
- Um caminho é uma sequência finita de vértices conectados por uma sequência de arestas;
- Um ciclo é um caminho fechado;
- O peso é uma medida atribuída a cada aresta. O peso de um caminho pode ser definido pelo número de saltos, onde cada aresta percorrida representa um salto;
- Um grafo é conexo se e somente se existe pelo menos um caminho interligando cada par de vértices origem e destino, caso contrário é um grafo desconexo. A Figura 1 apresenta um grafo conexo (a) e um grafo desconexo (b).

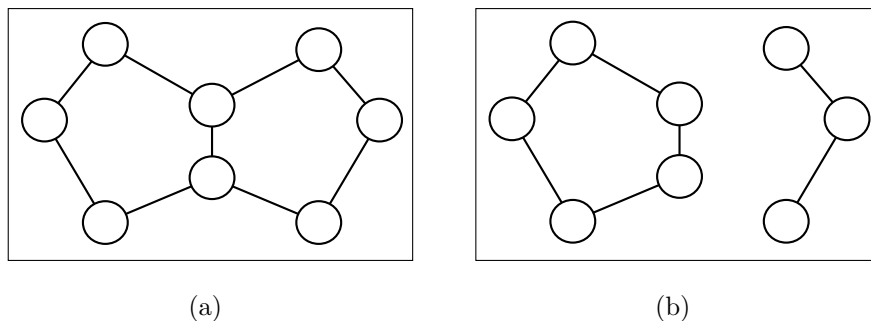


Figura 1 – (a) Grafo conexo. (b) Grafo desconexo.

- Um ponto de articulação é um vértice que ao ser retirado torna o grafo desconexo;
- Uma ponte é uma aresta que ao ser retirada torna o grafo desconexo;

- Em um grafo *2-conexo*, cada par origem e destino está interligado por pelo menos dois caminhos que não compartilham vértices. Uma rede *2-conexa* sobrevive a uma falha única em qualquer um de seus nodos. Para reconhecer uma rede *2-conexa*, basta identificar se existe algum ponto de articulação na rede;
- Em um grafo *2-aresta-conexo*, cada par origem e destino está interligado por pelo menos dois caminhos que não compartilham arestas. Uma rede *2-aresta-conexa* sobrevive a uma falha única em qualquer um de seus enlaces. Para reconhecer uma rede *2-aresta-conexa*, basta identificar se existe alguma ponte na rede;
- Uma árvore é um grafo conexo que não possui ciclos (acíclico);

1.2 Proteção em Redes Ópticas Elásticas

Em redes ópticas elásticas de telecomunicações, deve-se garantir uma rede confiável que sobreviva a alguns tipos de falhas, como de nodo ou de enlace, uma vez que uma única falha pode prejudicar significativamente o tráfego de dados e causar problemas de comunicação em conexões de diversos pares de nodos origem e destino [Pavan et al. 2015, Girolimetto e Pavan 2016]. Portanto, as redes devem ser sobreviventes, ou seja, mesmo em cenários de falhas devem ter soluções para que as conexões permaneçam disponíveis. Para isso, mecanismos de proteção são extremamente úteis pois apresentam caminhos alternativos para enviar o tráfego de pares origem e destino. Neste trabalho, as topologias consideradas são *2-aresta-conexas*, utilizando para cada par origem e destino, um caminho de trabalho e um caminho de *backup* (usado para proteção à falha única de enlace) e com peso de caminho medido em número de saltos.

Quando se utiliza esquemas de proteção, os caminhos de trabalho e de *backup* disjuntos por nodos ou por arestas devem ser definidos previamente, no estabelecimento da conexão, garantindo a recuperação em caso de falhas. Geralmente, o caminho de trabalho é responsável por carregar o tráfego enquanto a rede estiver operando sem falhas, e quando uma falha de nodo ou de enlace ocorre, o caminho de *backup* se responsabiliza pelo tráfego [Ramaswami, Sivarajan e Sasaki 2009]. Este esquema é conhecido como proteção de caminho dedicado 1:1. No entanto, ambos os caminhos podem ser usados simultaneamente, como é o caso do esquema de proteção de caminho dedicado 1+1 [Goscien et al. 2015].

Técnicas onde o caminho de *backup* é encontrado posteriormente a falha também existem, mas se encaixam em técnicas de restauração em vez de proteção [Ramamurthy, Sahasrabuddhe e Mukherjee 2003].

O que acontece é que em algumas topologias de rede, quando usamos esquemas de proteção e definimos o menor caminho como caminho de trabalho, nenhum caminho disjunto pode ser encontrado para um determinado caminho de *backup*, mesmo quando a

rede é *2-conexa*. A Figura 2 apresenta uma rede *2-conexa* que exemplifica essa situação, onde para um par com nodo origem O e nodo destino D um menor caminho é encontrado, mas um caminho de *backup* disjunto por arestas com relação ao primeiro não será possível.

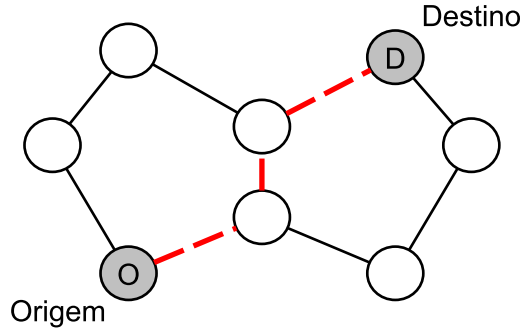


Figura 2 – Uma rede *2-conexa* para a qual, se o menor caminho entre os nodos O e D (o caminho de três saltos em linha tracejada vermelha) é escolhido como caminho de trabalho, não existe caminho disjunto para ser usado quando uma falha ocorrer.

Para encontrar dois caminhos disjuntos, existem estratégias e algoritmos de roteamento que podem ser usados. Normalmente o algoritmo *Suurballe e Tarjan* (*ST*), com dois caminhos disjuntos por nodos [Suurballe 1974] ou por arestas [Suurballe e Tarjan 1984] [Suurballe e Tarjan 1984] é utilizado [Kuipers 2012]. Entretanto, outros algoritmos que encontram caminhos disjuntos podem ser usados. Como algoritmos de K caminhos disjuntos por nodos ou por arestas [Li, McCormick e Simchi-Levi 1992], [Tholey 2012], [Park, Choi e Lim 2016], [Arora, Lee e Thurimella 2003] utilizado em redes móveis, [Zhang et al. 2014] utilizado em redes de sensores sem fio e versões de K caminhos disjuntos que encontram o número máximo de caminhos disjuntos por nodos ou por arestas como em [Dinneen, Kim e Nicolescu 2010] e em [Chan e Chin 1997]. Além disso, versões modificadas do *Dijkstra* também são utilizadas, como em [Dahshan 2010] e em [Dahshan 2013]. Neste trabalho usamos o algoritmo *ST* disjunto por arestas.

A finalidade do algoritmo *ST* é encontrar, para cada par de nodos origem e destino de uma rede, um par de caminhos disjuntos por arestas tais que a soma de seus comprimentos seja mínima. Nessa abordagem, o menor caminho do par é usado como caminho de trabalho e o outro como caminho de *backup*.

Em [Girolimetto e Pavan 2016] foi observado que, em certas topologias, pode haver mais de um par de caminhos disjuntos por arestas com o mesmo comprimento mínimo C interligando um mesmo par de nodos origem e destino. Por exemplo, na topologia de rede da Figura 3, o menor par de caminhos disjuntos por arestas que conecta o nodo O ao nodo D tem $C = 10$ saltos. Para este par, pode ser encontrado: *i*) um caminho de trabalho de 3 saltos e um caminho de *backup* de 7 saltos, $C = 3 + 7$, conforme mostrado na Figura 3(a);

ou *ii*) caminhos de trabalho e de *backup* de 5 saltos cada, $C = 5 + 5$, como mostrado na Figura 3(b).

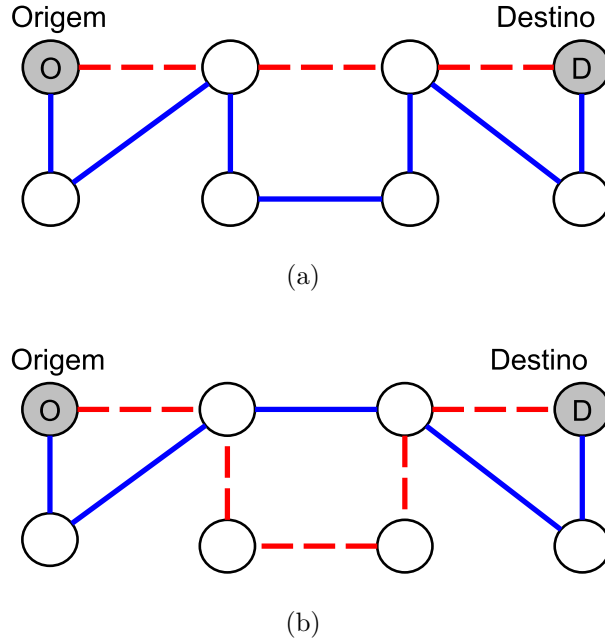


Figura 3 – Dois pares de caminhos disjuntos por arestas com comprimento mínimo igual a $C = 10$ interconectando o nodo origem O e o nodo destino D . Em (a), um caminho de trabalho com 3 saltos (linha pontilhada vermelha) e um caminho de *backup* com 7 saltos (linha cheia azul). Em (b), um caminho de trabalho e um outro de *backup* com 5 saltos cada.

A existência de mais de um par de caminhos com comprimento mínimo associado a um mesmo par de nodos origem e destino nos permite explorar diferentes estratégias para a escolha dos caminhos de trabalho e de *backup*, que podem ser utilizadas na definição de políticas de roteamento apropriadas para redes com diferentes tipos de serviço. Por exemplo, em caso de serviço sensível à latência, o ideal é que os caminhos de trabalho e de *backup* tenham o mesmo comprimento ou, pelo menos, que seus comprimentos tenham a menor diferença possível. Neste caso, deve-se adotar a estratégia de buscar o que é denominado neste trabalho como o par de caminhos *mais balanceado*. Essa exigência costuma ocorrer quando não há, na prática, distinção entre caminhos de trabalho e de *backup*: ambos os sinais são enviados simultaneamente para determinado destino e apenas o de melhor qualidade é utilizado. Em outros cenários, o ideal é que o caminho de trabalho seja o menor possível, e o caminho de *backup* é ativado apenas em caso de necessidade. Neste caso, a estratégia mais apropriada seria buscar o que é chamado neste trabalho de par de caminhos *menos balanceado*, de modo que o menor caminho do par seja utilizado como caminho de trabalho e o maior fique como *backup*. Ambas as estratégias são esquemas de proteção e são observadas em casos práticos como, por exemplo, em redes ópticas de transporte [Manchester, Bonenfant e Newton 1999].

Algoritmos de roteamento que definam caminhos de trabalho e de *backup* podem ser aplicados em esquemas de proteção, que geralmente são classificados como proteção dedicada e proteção compartilhada. Atualmente, simulações em esquemas de proteção são amplamente usadas para calcular métricas como a disponibilidade de conexões, taxas de falha e reparação, utilização de recursos e eficiência da proteção. Em [Zhang et al. 2003], [Fawaz et al. 2006] e [Song, Zhang e Mukherjee 2007] os autores apresentam diferentes abordagens sobre a métrica disponibilidade de conexões em diferentes esquemas de proteção e consideram uma taxa de falha proporcional ao comprimento da fibra. Em [Xu et al. 2013], a rede real *CERNET2* (*China Education and Research NETwork2*) foi analisada por aproximadamente um mês para identificar probabilidades de falhas. Nesse período, 240 falhas foram observadas. A taxa de falha de enlace segue uma função exponencial. Com base nos resultados de características de falha de enlaces, um novo esquema de proteção foi proposto. Em [Assi et al. 2002], esquemas de proteção de caminho dedicado e proteção de caminho compartilhado foram comparados quanto à probabilidade de bloqueio e o tempo de restauração sob a hipótese de falha. Em [Wang, Zeng e Zhao 2003], um conjunto de esquemas de proteção foi investigado e dois esquemas novos foram propostos. Um modelo de falha única de enlace foi definido e os esquemas de proteção foram comparados através de três métricas: uso de recursos, eficiência de proteção e interrupção de serviço.

A fim de explorar estratégias para a escolha dos caminhos de trabalho e *backup*, este trabalho propõe duas versões do algoritmo *ST* para encontrar, para uma determinada topologia, pares de caminhos de trabalho e de *backup* mais balanceados e menos balanceados para cada par de nodos. As versões propostas são denominadas: algoritmo de Suurballe e Tarjan Mais Balanceado (*Suurballe and Tarjan More Balanced - STMB*) e algoritmo de Suurballe e Tarjan Menos Balanceado (*Suurballe and Tarjan Less Balanced - STL B*). Para uma dada topologia *2-aresta-conexa*, o algoritmo *STMB* encontra o menor par de caminhos disjuntos por arestas para cada par de nodos origem e destino, com comprimentos dos caminhos de trabalho e de *backup* com a menor diferença possível. Ao contrário, o *STLB* mantém a maior diferença possível entre os comprimentos do caminho de trabalho e de *backup*.

Os algoritmos *STMB* e *STLB* foram testados e analisados através de um conjunto de 40 redes ópticas de telecomunicações reais [Pinto 2017]. O impacto destes algoritmos também foi investigado através *i)* do número de *transponders* (dispositivo de comunicação) necessários para encaminhar uma demanda de tráfego uniforme e *ii)* do coeficiente de proteção (a quantidade fracionada de capacidade adicional necessária para implementar o esquema de proteção de caminho dedicado) [Korotky 2004, Labourdette et al. 2005]. Além disso, foram realizadas simulações em cenários de tráfego dinâmico em redes ópticas elásticas utilizando o esquema de proteção de caminho dedicado 1:1 para duas redes ópticas de telecomunicações reais: “Arnes” e “Cernet”. Nas simulações, foram comparados o desempenho dos algoritmos propostos versus a carga de tráfego, o intervalo entre falhas

e a duração da falha. Com isso, foi obtida a taxa de bloqueio de requisições de conexão de cada algoritmo.

1.3 Objetivo Geral

O objetivo geral deste trabalho consiste em explorar a diversidade de pares de caminhos disjuntos por arestas com comprimento mínimo encontradas em um mesmo par de nodos origem e destino de uma topologia de rede. Para isso, duas soluções serão desenvolvidas no *framework ElasticO++* [Tessinari et al. 2016]. Estas soluções devem assegurar que os caminhos de trabalho e de *backup* obtenham comprimentos específicos, sendo uma com comprimentos com a menor diferença possível e outra com comprimentos com a maior diferença possível.

1.4 Objetivos Específicos

- Estudar a ferramenta *ElasticO++* e a linguagem de programação identificada no *framework*;
- Definir a melhor estratégia para desenvolver as soluções e implementá-las na ferramenta, de modo que encontrem dois caminhos disjuntos por arestas mais ou menos balanceados sempre que possível para uma topologia de rede *2-aresta-conexa*;
- Analisar como estas soluções se comportam, seu impacto e diferenças;
- Simular e comparar as duas soluções em um cenário de falhas de rede com proteção de caminho dedicado para identificar a utilização de recursos e a taxa de bloqueio.

1.5 Contribuições

Os resultados deste trabalho foram publicados em [Girolimetto et al. 2018] no Simpósio Brasileiro de Redes de Computadores (SBRC). O artigo [Girolimetto et al. 2018] apresenta duas versões do algoritmo *ST*. Estas versões encontram para uma dada topologia, os pares de caminhos de trabalho e de *backup* mais balanceado e menos balanceado chamados de Suurballe e Tarjan Mais Balanceado (*STMB*) e Suurballe e Tarjan Menos Balanceado (*STLB*). Estes algoritmos foram testados e analisados em um conjunto de 40 redes ópticas de telecomunicações reais. O impacto do *STMB* e do *STLB* foi investigado através do número de *transponders* e do coeficiente de proteção. Os resultados mostraram que dentre as redes testadas, será preciso até 29% mais *transponders* quando o *STMB* for utilizado e até 71% a mais de proteção extra quando se utiliza o *STLB*.

No *software ElasticO++* foram implementados os algoritmos: *Suurballe e Tarjan*, *Suurballe e Tarjan Mais Balanceado* e *Suurballe e Tarjan Menos Balanceado*. Além disso, também foi introduzido o esquema de proteção dedicada 1:1 e um módulo para gerenciar falhas.

1.6 Estrutura da Dissertação

Este trabalho é organizado da seguinte forma: o Capítulo 2 descreve as redes ópticas elásticas e destaca os esquemas de proteção; o Capítulo 3 descreve a ferramenta *ElasticO++* e os algoritmos base: *Suurballe e Tarjan* e *Yen*; os algoritmos propostos: *Suurballe e Tarjan Mais Balanceado* e *Suurballe e Tarjan Menos Balanceado* são apresentados no Capítulo 4; o Capítulo 5 descreve o passo a passo da metodologia utilizada; os resultados relativos ao comportamento, impacto e simulação de falhas utilizando os dois algoritmos são apresentados no Capítulo 6; e o Capítulo 7 traz as conclusões e sugestões para trabalhos futuros.

2 Redes Ópticas Elásticas

Este capítulo aborda as redes ópticas elásticas, uma evolução na tecnologia no campo das redes de telecomunicações. Suas características e diferenças com relação a redes ópticas *WDM* também são apresentadas. Além disso, um foco no mecanismo de sobrevivência de proteção é dissertado, destacando a proteção de caminho dedicado.

2.1 Introdução

O crescimento exponencial do tráfego devido a serviços multimídia e a constante mudança de provedores de conteúdo na *Internet* vem exigindo mudanças nas redes ópticas *WDM*. Tecnologias com mais eficiência, flexibilidade e escalabilidade são buscadas para atender as novas demandas.

A rede óptica tradicional baseada em Multiplexação por Divisão de Comprimento de Onda (*WDM*) divide o espectro em canais separados. Geralmente o espaçamento entre canais é de $50GHz$ ou $100GHz$, especificado pelo padrão da União Internacional de Telecomunicações (*ITU*) [Chatterjee, Sarma e Oki 2015]. Quando um tráfego é transmitido com baixa largura de banda e nenhum tráfego pode ser transmitido na lacuna de frequência não usada, uma grande parcela do espectro será desperdiçada [Chatterjee, Sarma e Oki 2015].

Para superar estas limitações, uma rede óptica elástica eficiente baseada na tecnologia Multiplexação por Divisão de Frequência Ortogonal (*OFDM*) foi apresentada [Zhang et al. 2013]. A tecnologia *OFDM* transmite um fluxo de dados de alta velocidade, dividindo-o em vários canais de dados paralelos. Os canais adjacentes podem se sobrepor ortogonalmente, aumentando a eficiência espectral de transmissão. Cada símbolo *OFDM* possui uma banda de guarda para evitar a interferência causada quando um símbolo *OFDM* cruza o limite de outro símbolo vizinho. Esta banda de guarda separa caminhos ópticos adjacentes em cada enlace [Christodoulopoulos, Tomkos e Varvarigos 2011]. Na Figura 4 são apresentadas bandas de guarda de espectro. Neste exemplo, sinais de diferentes caminhos ópticos são multiplexados no domínio da frequência [Christodoulopoulos, Tomkos e Varvarigos 2011].

As redes ópticas elásticas suportam altas taxas de dados flexíveis [Gerstel et al. 2012] e alocam espectro para caminhos ópticos de acordo com a largura de banda requisitada pelos clientes. O espectro é dividido em *slots* de frequência. Atualmente, a granularidade mínima de espectro que pode ser atribuída é $12,5 GHz$ [Patel et al. 2012, Tessinari et al. 2016]. Essa divisão do espectro também pode acomodar ambos “sub” e “super” canais

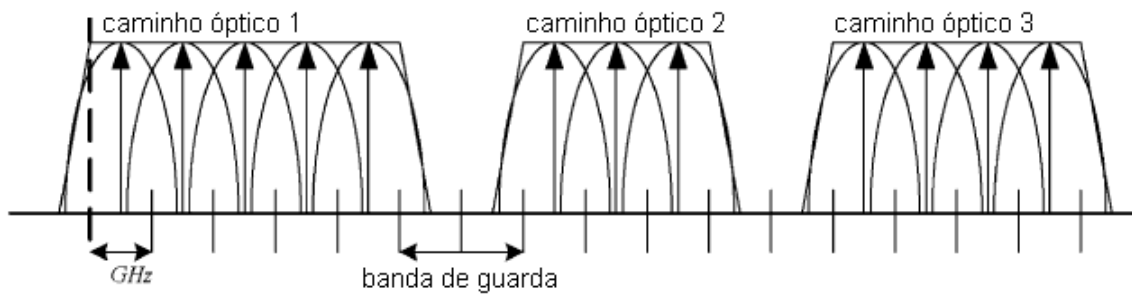


Figura 4 – Três caminhos ópticos separados por bandas de guarda. Adaptado de [Christodouloupoulos, Tomkos e Varvarigos 2011].

de comprimento de onda de acordo com cada requisição, garantindo transmissão de alta velocidade no enlace à medida que a demanda de tráfego aumenta.

Um caminho óptico pode se expandir e contrair de acordo com o volume de tráfego e a solicitação do usuário, se necessário. Como resultado, se obtém uma melhor eficiência da utilização da rede [Chatterjee, Sarma e Oki 2015, Jinno et al. 2009].

Uma rede óptica elástica pode operar em três modos de transporte: opaco, transparente ou translúcido. Uma rede configurada no modo de transporte opaco realiza a conversão óptico para elétrico para óptico (*OEO*) dos sinais no final de cada sistema de transmissão. Nesse tipo de rede, os sinais são regenerados em cada nodo, já que precisam ser convertidos para o domínio eletrônico. Por conta disso, um par de *transponders* é necessário para cada canal óptico abrangendo um enlace. A atualização da rede é custosa e para suportar taxas mais altas, novos nodos e *transponders* precisam ser implantados [Ramamurthy et al. 1999]. Já uma rede operando em um modo totalmente transparente, mantém o sinal de uma conexão no domínio óptico em todos os nodos intermediários, exceto nos nodos finais que o sinal não sofre conversão *OEO*. A capacidade de regeneração não está presente e é fácil atualizar a rede para taxas de bits mais altas [Bouillet et al. 2007]. E por fim, uma rede configurada no modo translúcido, que é uma junção do modo opaco com o modo transparente, aproveita as vantagens destes modos de transporte. Nesta arquitetura, os sinais trafegam pela rede “o máximo possível” no domínio óptico. O sinal pode ser regenerado várias vezes na rede antes de atingir seu destino e a atualização da rede é fácil [Ramamurthy et al. 1999]. Neste trabalho foi utilizado o modo de transporte opaco.

2.2 Arquitetura das Redes Ópticas Elásticas

A arquitetura típica da rede óptica elástica consiste do *Transponder* de Largura de Banda Flexível (*BVT*) e do Comutador Óptico de Largura de Banda Flexível (*BV-WXC*), apresentados na Figura 5 [Chatterjee, Sarma e Oki 2015]. A arquitetura apresentada na

Figura 5 é simples. Portanto, o número de *transponders* pode ser diferente dependendo do modo de transporte utilizado.

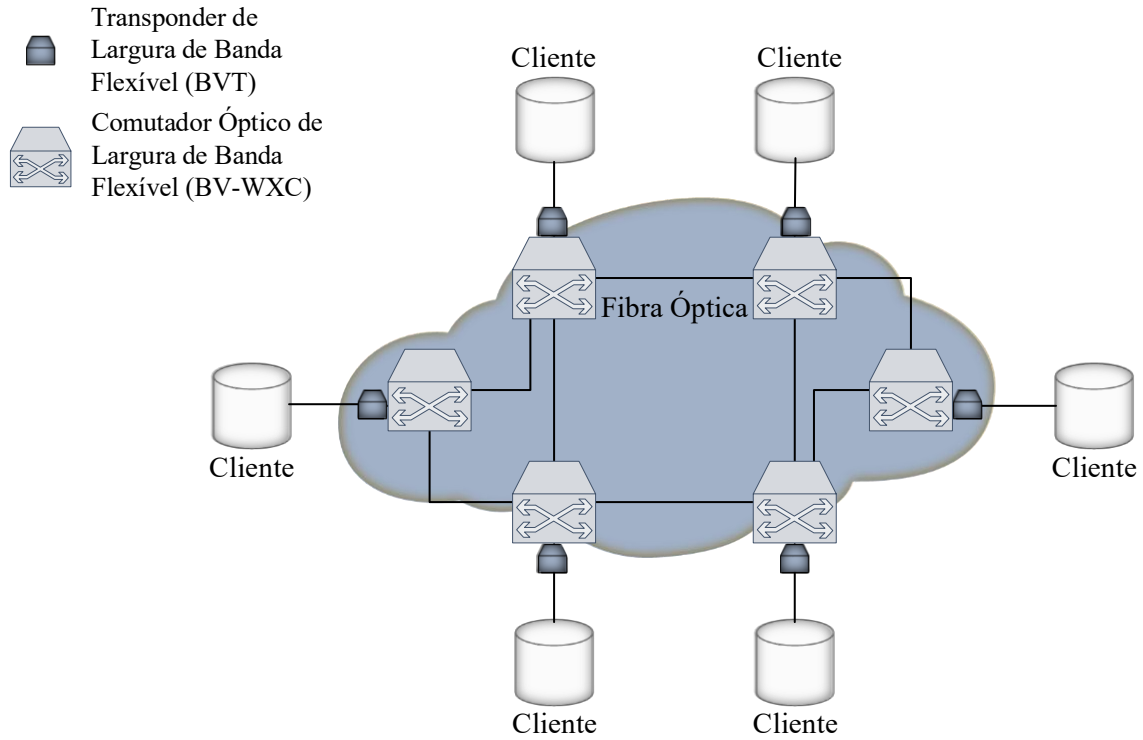


Figura 5 – Arquitetura do enlace e do nó numa rede óptica elástica, apresentando o *BVT* e o *BV – WXC*.

Os *BVTs* estão conectados aos *BV – WXC*s e são usados para ajustar a largura de banda, alocando espectro suficiente para acomodar cada demanda. Além disso, suportam a transmissão de alta velocidade usando formatos de modulação espectralmente eficientes como a Modulação de Amplitude em 16-Quadratura (16 – *QAM*) e 64 – *QAM* usado para caminhos ópticos de menor distância. Os caminhos ópticos de maior distância usam formatos de modulação menos eficientes como a Modulação por Deslocamento de Fase em Quadratura (*QPSK*) ou Modulação por Deslocamento de Fase Binária (*BPSK*) [Chatterjee, Sarma e Oki 2015]. No entanto, quando um *BVT* de alta velocidade é operado a uma taxa menor do que a máxima devido ao alcance necessário ou devido às deficiências no caminho óptico, parte da capacidade do *BVT* é desperdiçada. Para resolver esse problema, um *Sliceable Transponder de Largura de Banda Flexível (SBVT)* que oferece maior flexibilidade foi apresentado. Um *SBVT* aloca sua capacidade em um ou vários fluxos ópticos que são transmitidos para um ou vários destinos. Os fluxos ópticos podem ser agregados ou divididos com base nas necessidades de tráfego [Chatterjee, Sarma e Oki 2015].

O *BV – WXC* aloca um comutador óptico de tamanho apropriado com a largura de banda do espectro correspondente estabelecendo um caminho óptico do nó origem

ao nodo destino, acomodando os recursos espectrais estabelecidos pelos *BVTs* [Jinno et al. 2009]. Quando os *BVTs* aumentam a taxa do tráfego, cada *BV – WXC* na rota deve expandir sua janela de comutação, permitindo assim uma taxa de dados variável em cada caminho óptico [Zhang et al. 2013]. Em um *BV – WXC*, Comutadores Seletivos de Espectro de Largura de Banda Flexível (*BVSSSs*) são usados para fornecer funcionalidades de roteamento e de *add-drop* para sinais locais [Chatterjee, Sarma e Oki 2015].

2.2.1 Arquitetura de Nodos

As principais arquiteturas de nodos das redes ópticas elásticas são: “Transmissão e Seleção”, “Roteamento de Espectro”, “Comutar e Selecionar com Funcionalidade Dinâmica” e “Arquitetura Sob Demanda”. A seguir estas arquiteturas são apresentadas e comparadas em termos de flexibilidade. No contexto de arquitetura de nodos, a flexibilidade é a capacidade do sistema de transportar sinais da origem para o destino ao longo de diferentes rotas. Esta flexibilidade é necessária para garantir a sobrevivência a falhas, direcionando sinais de caminhos de trabalho para seus respectivos caminhos de *backup*.

A arquitetura de nodo “Transmissão e Seleção” usa o *BVSSS* para filtrar sinais de entrada e selecionar o sinal requisitado para a saída [Amaya, Zervas e Simeonidou 2013]. Esta arquitetura permite a adição de um ou mais canais de comprimento de onda a um sinal existente de múltiplos comprimentos de onda automaticamente. As principais desvantagens são: (i) requer sincronização e ajuste rápido, (ii) não pode suportar reutilização de comprimento de onda e, portanto, é necessário um grande número de canais de comprimento de onda, (iii) a energia é dividida entre vários nodos, portanto não pode ser usado para comunicação de longa distância e (iv) não suporta desfragmentação. Esta arquitetura é utilizada principalmente em redes locais de alta velocidade e redes de área metropolitana [Chatterjee, Sarma e Oki 2015].

A arquitetura de nodo “Roteamento de Espectro” deve superar os problemas com a arquitetura de nodo “Transmissão e Seleção”. Nesta arquitetura, o *BVSSS* controla funcionalidades de comutação e filtragem [Amaya, Zervas e Simeonidou 2013]. *BVSSSs* adicionais são requisitados na entrada, o que a torna mais cara. Além disso, a desfragmentação do espectro também é difícil de implementar. [Chatterjee, Sarma e Oki 2015].

A arquitetura de nodo “Comutar e Selecionar com Funcionalidade Dinâmica” suporta funcionalidades dinâmicas como desfragmentação de espectro, multiplexação de tempo e regeneração. Nessa arquitetura, cópias dos sinais da entrada são direcionadas para um *BVSSS* específico ou para um módulo que fornece funcionalidades dinâmicas [Amaya, Zervas e Simeonidou 2013, Chatterjee, Sarma e Oki 2015].

A arquitetura de nodo “Arquitetura Sob Demanda” consiste de um comutador óptico com grande número de portas de entrada e saída conectado a vários módulos de processa-

mento como *BVSSs*. Esta arquitetura oferece maior flexibilidade do que as anteriores, pois os componentes podem ser interconectados juntos de maneira arbitrária. Além disso, proporciona ganhos consideráveis em termos de escalabilidade e resiliência [Chatterjee, Sarma e Oki 2015].

A arquitetura de nodo com alta flexibilidade no roteamento é a “Arquitetura Sob Demanda”, com média flexibilidade no roteamento é a “Comutar e Selecionar com Funcionalidade Dinâmica” e com nenhuma flexibilidade no roteamento são a “Transmissão e Seleção” e o “Roteamento de Espectro” [Chatterjee, Sarma e Oki 2015].

2.3 Roteamento e Atribuição de Espectro

Um problema essencial para o gerenciamento do espectro em *EON's* é o Roteamento e Atribuição de Espectro (*RSA*). Este problema consiste em (i) calcular uma série de rotas candidatas para cada par de nodos origem-destino e (ii) alocar um conjunto de *slots* contíguos e contínuos, se existirem, no espectro usando uma política de alocação de espectro para atribuir um caminho viável na conexão solicitada [Christodoulopoulos, Tomkos e Varvarigos 2011, Gerstel et al. 2012].

O problema *RSA* é equivalente ao Problema de Roteamento e Alocação de Comprimento de Onda (*RWA*) em redes ópticas baseadas em *WDM*. O *RWA* estabelece caminhos ópticos para cada solicitação de conexão selecionando uma rota apropriada e atribuindo um comprimento de onda desejado. O mesmo comprimento de onda deve ser usado em todos os enlaces do caminho origem-destino de uma conexão. Esta propriedade é conhecida como restrição de continuidade do comprimento de onda [Chatterjee, Sarma e Oki 2015]. A diferença entre estes problemas é que a rede óptica elástica oferece alocação de espectro flexível para atender as taxas de dados solicitadas e um conjunto de *slots* contíguos é alocado para uma conexão em vez do comprimento de onda definido pelo *RWA*. As restrições referentes ao *RSA* são de (i) contiguidade: se uma demanda requer x unidades de espectro, então x *slots* contíguos devem ser alocados; e (ii) continuidade: os mesmos x *slots* contíguos devem ser alocados numa mesma posição em todos os enlaces do caminho origem-destino de uma conexão (semelhante à restrição do *RWA*) [Talebi et al. 2014].

Considerando a rede apresentada na Figura 6, um exemplo sobre os conceitos das restrições de contiguidade e continuidade do *RSA* é abordado. Vamos supor uma requisição de conexão que requer uma taxa de bit equivalente a dois *slots* e sem banda de guarda, para uma origem v_1 e um destino v_4 . Então, dado o caminho $v_1 \leftrightarrow v_2 \leftrightarrow v_4$ é verificado se as restrições podem ser atendidas. O enlace $v_1 \leftrightarrow v_2$ possui os *slots* contíguos disponíveis: 1, 2 e 7, 8. Um destes pares poderia ser escolhido pois a restrição de contiguidade está garantida. Porém, é verificado que no enlace $v_2 \leftrightarrow v_4$, os *slots* contíguos disponíveis são: 5, 6, 7 e

9, 10. Neste caso, não é possível respeitar a restrição de continuidade. Já se utilizarmos o caminho $v_1 \leftrightarrow v_3 \leftrightarrow v_4$, ambas as restrições são possíveis utilizando os *slots* contíguos e contínuos disponíveis: 8, 9 dos enlaces $v_1 \leftrightarrow v_3$ e $v_3 \leftrightarrow v_4$.

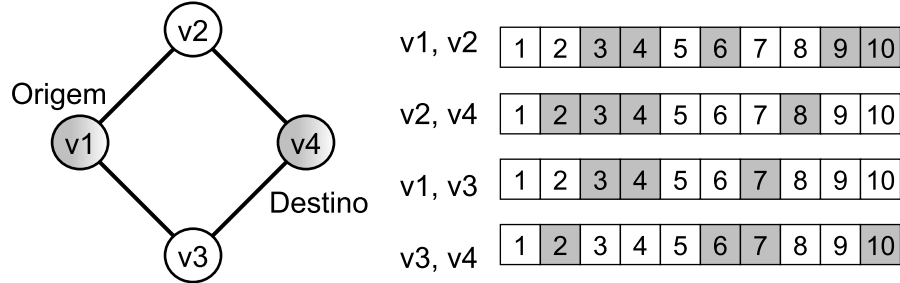


Figura 6 – Exemplo referente às restrições de contiguidade e continuidade do *RSA*. *Slots* disponíveis estão representados pela cor branca e *slots* ocupados estão representados pela cor cinza.

Uma vez que o desempenho de uma rede depende não apenas dos seus recursos físicos (por exemplo: *transponders*, enlaces, largura espectral utilizável, comutadores ópticos, etc.), mas também sobre como é controlada, o objetivo de um algoritmo *RSA* é alcançar o melhor desempenho possível com os limites das restrições físicas [Talebi et al. 2014].

O *RSA* possui variantes, classificados em duas versões: *RSA offline*, onde as demandas de tráfego são conhecidas antecipadamente, e *RSA online*, onde uma sequência de solicitações dos clientes chega de forma aleatória [Talebi et al. 2014]. O *RSA offline* surge sempre que os padrões de tráfego na rede são razoavelmente bem conhecidos. Os caminhos são calculados e ordenados previamente. Já o *RSA online* é empregado em um cenário de tráfego dinâmico, onde os caminhos ópticos são configurados conforme necessário. Dependendo do estado da rede no momento de uma solicitação, os recursos espectrais disponíveis podem ou não ser suficientes para estabelecer uma conexão entre o par de nodos origem e destino correspondente. Se um pedido de conexão não puder ser aceito por falta de recursos, ele é bloqueado. Portanto, a probabilidade de bloqueio dos pedidos de conexão surge como a métrica de desempenho de interesse em um cenário *RSA online* [Talebi et al. 2014].

O problema *RSA* é redutível ao problema *RWA* que é NP-difícil. Para qualquer conexão, se o *RWA* requer um comprimento de onda ao longo do caminho óptico, é equivalente dizer que o *RSA* requer um *slot* de um espectro ao longo do caminho óptico. A redução está em tempo polinomial. O problema *RWA* tem uma solução se e somente se o problema *RSA* construído tiver uma solução [Chatterjee, Sarma e Oki 2015]. Portanto, pode-se dizer que o problema *RSA* é um problema NP-difícil [Christodouloupoulos, Tomkos e Varvarigos 2011].

2.4 Fragmentação

A configuração dinâmica de conexões e a flexibilidade do espectro nas redes ópticas elásticas, podem gerar o problema de fragmentação de banda. Os *slots* disponíveis que estão isolados, desalinhados ou descontíguos se tornam inaptos para a alocação de futuras conexões. Portanto, se nenhum conjunto de *slots* disponíveis puder satisfazer a demanda de largura de banda necessária, o pedido de conexão é rejeitado. Assim, ocorre um bloqueio de requisição, e operadores de rede devem reconfigurar periodicamente os caminhos ópticos e os *slots* do espectro, realizando uma desfragmentação da rede [Chatterjee, Sarma e Oki 2015].

A fragmentação pode ser horizontal e/ou vertical. A restrição de continuidade do espectro causa a fragmentação horizontal, onde o mesmo bloco de espectro pode não estar disponível ao longo de enlaces sucessivos de um caminho, apesar do fato de cada enlace ter uma largura de banda suficiente para uma solicitação. Já as taxas de dados variáveis juntamente com a restrição de contiguidade do espectro, por outro lado, são a causa da fragmentação vertical, uma situação em que, os recursos espectrais em um único enlace são fragmentados em pequenos blocos não contíguos, que não podem ser alocados a uma única demanda grande [Talebi et al. 2014].

2.5 Mecanismos de Sobrevivência

Redes ópticas elásticas têm a capacidade de suportar taxas de dados individuais de $40 - 1000 \text{ Gb/s}$, e também agregam a taxa de transferência por fibra a aproximadamente $10 - 100 \text{ Tb/s}$. Portanto, a falha de um componente de rede, como um enlace ou nodo de rede, pode interromper comunicações levando a uma grande perda de dados. Para contornar este problema, mecanismos de sobrevivência podem ser aplicados para proteção da rede.

Semelhante às redes ópticas baseadas em *WDM*, os mecanismos de sobrevivência para redes ópticas elásticas podem ser classificados em duas grandes categorias: proteção e restauração (Figura 7) [Chatterjee, Sarma e Oki 2015]. As técnicas de proteção utilizam caminhos de *backup* previamente calculados que transportam o tráfego antes ou após a ocorrência de falha. Esta escolha depende do esquema de proteção definido. A proteção é classificada em proteção de caminho e proteção de enlace, sendo a primeira dividida em proteção de caminho dedicada e proteção de caminho compartilhada. Nas técnicas de restauração, os caminhos de *backup* são calculados dinamicamente com base em informações do estado da rede após a ocorrência de falha. A restauração é classificada em restauração de enlace, restauração de sub-caminho e restauração de caminho. Aqui, destacaremos a técnica de proteção de caminho dedicado, dividida em 1:1 e 1+1.

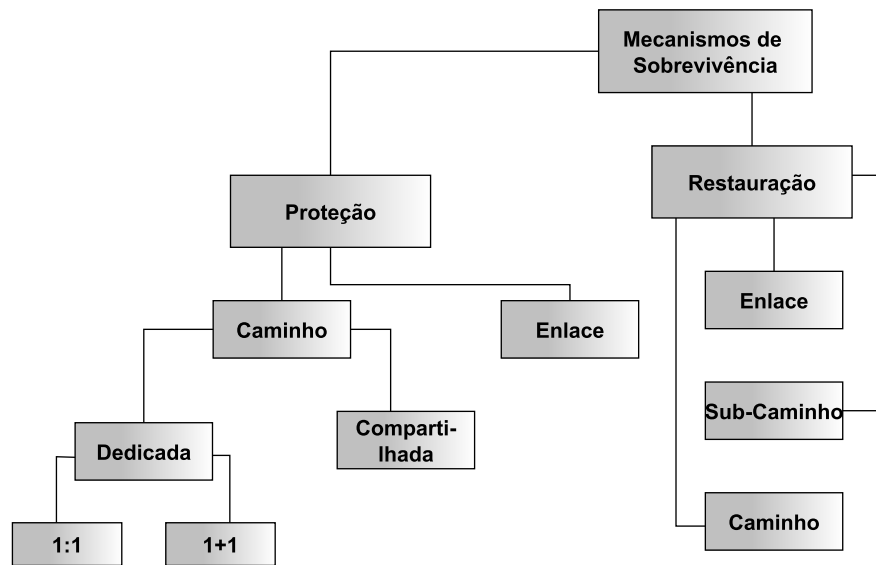


Figura 7 – Mecanismos de sobrevivência em redes [Saini e Garg 2013].

2.5.1 Esquemas de Proteção

Esquemas de proteção são classificados pelo roteamento utilizado: proteção do enlace ou proteção do caminho; ou pelo tipo de compartilhamento de recursos: proteção dedicada ou proteção compartilhada [Ramamurthy, Sahasrabuddhe e Mukherjee 2003].

Estes esquemas também podem ser reversíveis ou não reversíveis. Num esquema não reversível, quando uma falha ocorre no caminho de trabalho, o tráfego é trocado para o caminho de *backup* e permanece lá. Num esquema reversível, o tráfego é trocado de volta para o caminho de trabalho quando a falha é reparada [Ramaswami, Sivarajan e Sasaki 2009].

2.5.1.1 Proteção de Enlace e de Caminho

Na proteção de enlace quando uma falha acontecer, todas as conexões que atravessam o enlace com falha serão reencaminhadas para os enlaces ao redor desse enlace.

Na proteção de caminho, é reservado um caminho de *backup* totalmente disjunto por enlaces ou nodos com relação ao caminho de trabalho para cada par de nodos origem e destino de uma topologia de rede. No instante em que ocorrer uma falha, o caminho de *backup* é utilizado [Ilyas e Mouftah 2003].

2.5.1.2 Proteção Dedicada

Na proteção de enlace dedicada, no momento da configuração da conexão, para cada enlace do caminho de trabalho, um caminho de *backup* é reservado em torno desse enlace e é dedicado a essa conexão [Ilyas e Mouftah 2003].

Na proteção de caminho dedicado, um caminho de trabalho e um de *backup* são definidos para cada par de nodos da rede e os recursos não podem ser compartilhados com outros pares. Este esquema de proteção pode ser reversível ou não reversível. A proteção de caminho dedicado pode ser implementada de duas maneiras diferentes, conhecidas como 1:1 e 1+1 [Lang, Rekhter e Papadimitriou 2007, Berger et al. 2007].

Na proteção 1:1, os recursos para o caminho de *backup* são reservados apenas para garantir a recuperação quando uma falha ocorrer. O tráfego é transmitido apenas em um caminho por vez. O caminho de trabalho recebe o tráfego regular, enquanto o caminho de *backup* pode receber tráfego de baixa prioridade. Quando ocorrer uma falha, o tráfego enviado ao caminho de trabalho é trocado para o caminho de *backup*. O tráfego de baixa prioridade é interrompido do caminho de *backup* antes de ser usado pelo tráfego regular para evitar erros de conexões entre os tráfegos.

Na proteção 1+1, os recursos reservados para o caminho de *backup* são utilizados o tempo todo juntamente com os recursos do caminho de trabalho pois a transmissão do tráfego ocorre em ambos ao mesmo tempo. No destino, o caminho com melhor sinal é selecionado. Após uma falha, o tráfego permanece em um caminho e a conexão permanece ativa [Ramaswami, Sivarajan e Sasaki 2009].

Nos esquemas de proteção de caminho dedicado, a proteção 1+1 é bastante eficiente em tempo de recuperação, mas custosa em termos de uso de largura de banda. Já a técnica 1:1 requer maior tempo de recuperação [Goscien et al. 2015].

2.5.1.3 Proteção Compartilhada

Na proteção de enlace compartilhado, os recursos de um enlace que pertençam a um caminho de *backup* podem ser compartilhados com outros caminhos de *backup*. Como resultado, eles são multiplexados entre diferentes cenários de falha [Ilyas e Mouftah 2003].

Na proteção de caminho compartilhado, os recursos reservados nos caminhos de *backup* podem ser compartilhados com outros caminhos de *backup*, desde que não falhem simultaneamente. Quando não há falhas, os caminhos de *backup* estão aptos para transportar tráfego de baixa prioridade [Ramaswami, Sivarajan e Sasaki 2009].

Os esquemas de proteção compartilhada geralmente são reversíveis. Isso porque o caminho de *backup* deve ser liberado o mais rapidamente possível após a falha ter sido reparada, pra que este caminho possa proteger outras conexões em caso de nova falha [Ramaswami, Sivarajan e Sasaki 2009].

2.6 Transponders Ópticos e o Coeficiente de Proteção na Proteção de Caminho Dedicado

O dispositivo de comunicação *transponder* pode receber, amplificar e retransmitir um sinal em uma frequência diferente do sinal de entrada. Os *transponders* ópticos fazem parte da arquitetura da rede óptica elástica, apresentada na Seção 2.2 e possuem suporte a múltiplas taxas de bits (de 10 Gb/s a 1 Tb/s) e formato de modulação ajustável para cada conexão. Deste modo, este dispositivo pode evitar o uso de regeneradores (que compensa distorções do sinal em sistemas de longa distância) ao obter um formato eficiente para a distância necessária através da configuração do *transponder* [Sambo et al. 2015].

Para implementar proteção de caminho dedicado em redes ópticas que operam no modo de transporte opaco cada canal óptico abrangendo um enlace requer um par de *transponders*. Portanto, o número médio de *transponders* ópticos por enlace para os caminhos de trabalho é dado por [Pavan et al. 2015]:

$$T(W) = 2 \frac{\langle d \rangle \langle h^w \rangle}{\langle \delta \rangle}, \quad (1)$$

onde o 2 representa o par de *transponders*, $\langle d \rangle$ é o número médio de demandas de tráfego assumindo um modelo de demanda uniforme, em que cada nodo exige uma demanda para todos os outros nodos, $\langle h^w \rangle$ é o número médio de saltos considerando apenas os caminhos de trabalho, e $\langle \delta \rangle$ é o grau nodal médio da rede.

Similarmente, a média do número de *transponders* ópticos por enlace para os caminhos de *backup* é dada por:

$$T(B) = 2 \frac{\langle d \rangle \langle h^b \rangle}{\langle \delta \rangle}, \quad (2)$$

onde $\langle h^b \rangle$ é a média do número de saltos considerando apenas caminhos de *backup*.

Os caminhos de *backup* podem proteger a rede de falhas. Portanto, o custo extra necessário para proteger a rede de falhas é avaliado através da relação entre a quantidade total de recursos reservados para caminhos de *backup* e a quantidade total de recursos reservados para caminhos de trabalho [Korotky 2004, Labourdette et al. 2005]. Esta relação é referida como coeficiente de proteção (k_p) [Pavan et al. 2015] ou como redundância de capacidade extra [Shen, Guo e Bose 2016]. O coeficiente de proteção para o caminho de proteção dedicado é calculado desta forma [Pavan et al. 2015]:

$$\langle k_p \rangle = \frac{\langle h^b \rangle}{\langle h^w \rangle}. \quad (3)$$

Note que, uma vez que $\langle h^b \rangle$ é sempre maior ou igual a $\langle h^w \rangle$, $\langle k_p \rangle$ é sempre maior ou igual a 1. Portanto, quanto maior o $\langle k_p \rangle$, maior o custo para proteger a rede contra falhas.

Tanto o número de *transponders* quanto o coeficiente de proteção dependem diretamente dos comprimentos dos caminhos de trabalho e de *backup*. Assim, diferentes esquemas de proteção de caminho dedicado podem se beneficiar de diferentes maneiras ao definir esses caminhos.

2.7 Conclusão

Este capítulo apresentou as redes ópticas elásticas, determinando suas características, arquitetura e problemas de roteamento, atribuição de espectro e fragmentação de rede. Além disso, mecanismos de sobrevivência foram discutidos com uma ênfase em esquemas de proteção, abordando proteção de enlace e de caminho, proteção dedicada e compartilhada. Em adição, o cálculo de *transponders* ópticos e o coeficiente de proteção no esquema de proteção de caminho dedicado foram estabelecidos.

3 Ferramentas e Algoritmos

Neste capítulo é apresentado o *software* simulador de redes ópticas elásticas *ElasticO++* e em seguida os algoritmos de roteamento *Suurballe e Tarjan* e *Yen*. Estas foram ferramentas principais para o desenvolvimento dos algoritmos propostos neste trabalho.

O *ElasticO++* é um *framework* que permite testes em roteamento, modulação, atribuição de espectro, desfragmentação, parâmetros físicos da rede como atenuação, potência de entrada e de saída, entre outros e topologias, proporcionando o desenvolvimento de novos algoritmos e técnicas para redes ópticas elásticas. Nesta ferramenta existem alguns algoritmos de roteamento, como o algoritmo *Suurballe e Tarjan (ST)*, que busca dois caminhos disjuntos por nodos [Suurballe 1974] ou por arestas [Suurballe e Tarjan 1984], este último adicionado no *ElasticO++* a partir deste trabalho e o algoritmo de *Yen* [Yen 1970] que busca K menores caminhos, implementado anteriormente a este trabalho.

3.1 ElasticO++

O *ElasticO++* é uma estrutura de simulação para redes ópticas elásticas [Tessinari et al. 2016], projetada a partir de um simulador de eventos discreto chamado *OMNeT++* [Varga 2017].

O *OMNeT++* é um dos simuladores mais populares na área de pesquisa de redes de comunicação. Ele pode ser utilizado na modelagem de redes de comunicação, multiprocessadores e outros sistemas distribuídos ou paralelos. Sendo um *software* de código aberto, é livre para uso sem fins lucrativos.

A estrutura do *software ElasticO++* fornece um conjunto de instrumentos permitindo a rápida implementação, testes e análise de algoritmos em simulações de redes ópticas elásticas. Além disso, possibilita a execução de simulações com grandes quantidades de dados e suporte a gráficos e geração de estatísticas. Em sua versão atual (sem a parte adicionada por este trabalho), o *framework* vem com doze algoritmos tradicionais já implementados que podem ser usados de forma autônoma ou em combinação com novos, são eles: *Dijkstra*, *K-Shortest Paths*, *EON DP-QPSK*, *WDM DP-QPSK*, *Spectrum Sharing*, *Zone Based*, *First Fit*, *Random Fit*, *Make-Before-Break*, *Spectrum Compactness*, *Push-Pull* e *Always ON*. No momento o *software* suporta o problema Roteamento, Modulação e Atribuição de Espectro (*RMSA*) em cenários dinâmicos, mas pode ser estendido para cenários estáticos. A flexibilidade oferecida pela ferramenta permite que a academia e a indústria desenvolvam e avaliem novos algoritmos e técnicas para redes ópticas elásticas [Tessinari et al. 2016, Tessinari et al. 2015, Tessinari, Colle e Garcia 2017].

3.1.1 Arquitetura do ElasticO++

A estrutura da ferramenta *ElasticO++* é apresentada na Figura 8 sendo dividida em três partes: Ferramentas de *Script*, Equipamento e Controlador.

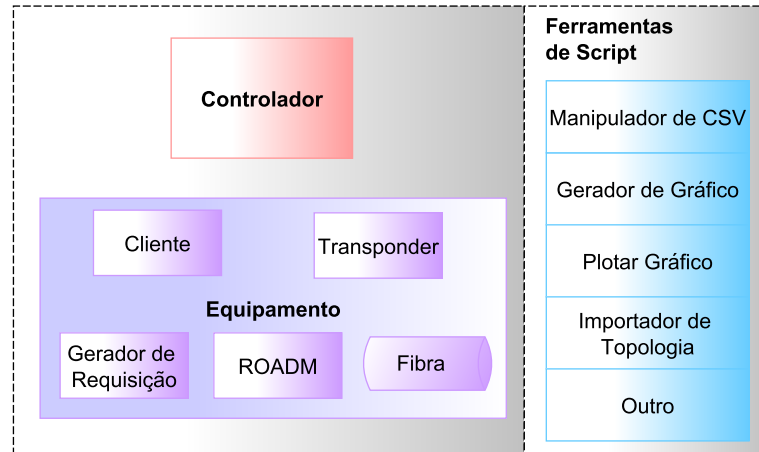


Figura 8 – Arquitetura da ferramenta *ElasticO++*. Adaptado de [Tessinari et al. 2016].

Em “Ferramentas de *Script*” se encontra um conjunto de *scripts* criados para automatizar tarefas como o gerenciamento de dados e a geração de gráficos. Nesta parte, existe o manipulador de *CSV*, *script* de gerador e plote de gráfico, importação de topologias, entre outros. Atualmente, o *framework* oferece suporte a três “formatos de resultados de gráfico”: *Microsoft Excel*, *OriginLab Origin* e *Matplotlib*.

O “Equipamento” possui a implementação de equipamentos convencionais encontrados em uma rede óptica, como *transponders*, Multiplexadores Ópticos *Add-Drop* Reconfiguráveis (*ROADM*) e fibra óptica. Além disso, configura diferentes padrões de tráfego através de um “Gerador de Requisição”.

Por último, o “Controlador”, considerado a parte mais importante do *ElasticO++*. O “Controlador” é composto de vários módulos organizados em três subdivisões: configuração, armazenamento e execução, conforme Figura 9. O módulo de configuração inicia todos os serviços de outros módulos com o “Escalonador” e detecta a topologia e recursos da rede com o “Gerente de Topologia”. Já no módulo de armazenamento encontra-se a “Tabela de Topologia” a qual possui informações da matriz de adjacência e matriz de enlaces, a “Tabela de Rotas” que gerencia rotas já encontradas de cada par de nodos origem e destino e a “Tabela de Conexão” que mantém as informações sobre conexões ativas. Nesta divisão também encontra-se uma classe chamada de “Enlace”, onde estão todas as características de um enlace, como os recursos alocados. Por último, o módulo de execução que recebe as requisições dos clientes no “Controle de Admissão”. O “Controle de Admissão” faz a interface entre o equipamento e outros módulos de controle: “Gerente de Recursos”, “Gerente de Alocação”, “Gerente de Log” e “Gerente de Desfragmentação”.

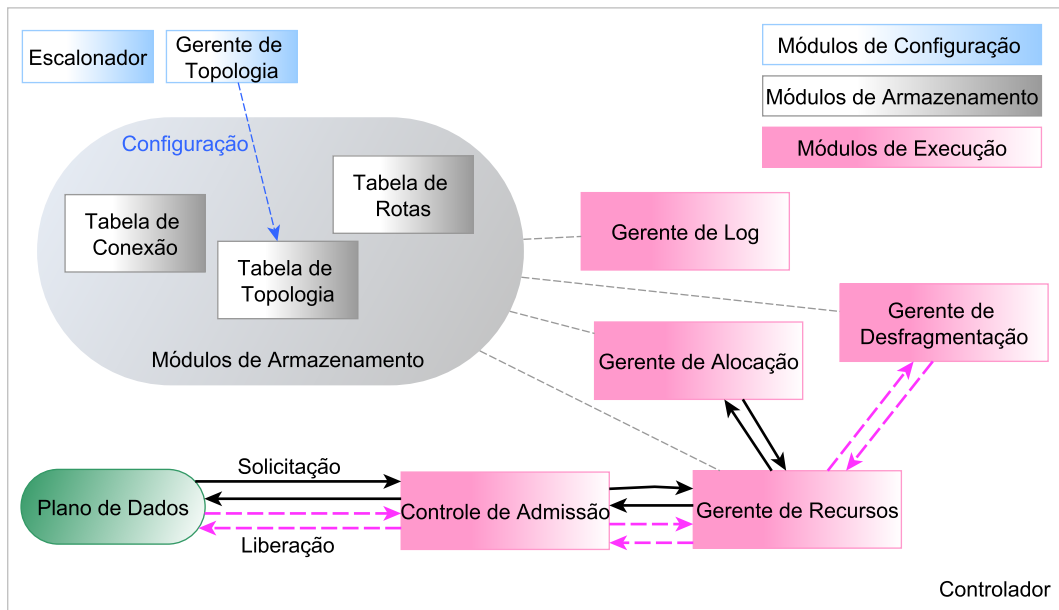


Figura 9 – Módulos de controle da ferramenta *ElasticO++*. Adaptado de [Tessinari et al. 2016].

O “Controle de Admissão” encaminha a “solicitação” para o módulo de “Gerente de Recursos” que é responsável por coordenar o “Gerente de Alocação” e o “Gerente de Desfragmentação”, e também por criar e remover conexões da “Tabela de Conexão”, e chamar o “Gerente de Log” para desenvolver *logs* de simulação. O “Gerente de Recursos” envia a “solicitação” para o “Gerente de Alocação” que é responsável pela montagem, gerenciamento e execução de algoritmos de alocação e executa o *RMSA* para verificar se existem recursos para a “solicitação”, retornando uma resposta ao “Gerente de Recursos” que atualiza as estatísticas de acordo com os resultados. Para o processo de alocação de conexões, existem quatro estruturas de dados diferentes usadas pelos módulos de controle: “Requisição de Alocação”, “Restrição de Alocação”, “Resultado de Alocação” e “Configuração de Alocação”. Cada estrutura possui características específicas referentes à alocação de conexões. O módulo “Gerente de Log” é responsável por manter estatísticas e relatórios de *logs* e é capaz de gerar um resultado de simulação em versão *CSV*. Por fim, uma resposta é retornada ao cliente solicitante, informando um número identificando a nova conexão ou somente a rejeição da mesma.

O processo de “liberação” funciona de modo semelhante ao processo de “solicitação”. A exceção é que o “Gerente de Recursos” não chama o “Gerente de Alocação”. Após a conexão ser encerrada, opcionalmente é possível chamar o “Gerente de Desfragmentação”, que utiliza técnicas para contornar o problema da fragmentação dos *slots*. Na sequência, o “Gerente de Log” é chamado novamente, as estatísticas são gravadas e uma resposta é retornada ao cliente solicitante informando o término da conexão solicitada.

Com relação a arquitetura do *ElasticO++* e módulos relevantes para este trabalho,

no “Controlador” todos os módulos são usados, com exceção do módulo “Gerente de Desfragmentação”. No “Equipamento”, os módulos “Cliente” e “Gerador de Requisição” são utilizados para a definição dos nodos origem e destino e pedidos de conexão. E por último nas “Ferramentas de *Script*”, o “Gerador de Gráfico” e “Plotar Gráfico” são usados para apresentar resultados das simulações e o “Importador de Topologia” para importar novas topologias de redes para serem utilizadas nas simulações. Além disso, foi adicionado um novo módulo ao “Equipamento”, chamado de “Gerador de Falhas”. No módulo “Gerador de Falhas” ocorre o controle de lançamento de falhas em enlaces aleatórios, e a ativação ou desativação de falhas de enlace. Ele se comunica com o “Gerador de Requisição” e com o “Cliente” para chegar na “Tabela de Rotas” e verificar caminhos afetados por falhas.

3.1.2 Processo de Requisição e Alocação de Recursos

Neste momento, apresenta-se o processo de requisição e alocação de recursos espectrais do *software ElasticO++*.

Dada uma topologia de rede, o algoritmo de roteamento identifica os caminhos de trabalho e de *backup* para todos os pares de nodos origem e destino. Este é um processo *offline* que gera uma tabela de roteamento. Após, seguindo o modelo de Erlang [Mandelbaum e Zeltyn 2004], requisições chegarão de acordo com um processo de *Poisson* para o qual o tempo de espera é distribuído exponencialmente (com uma média normalizada de $1/\mu = 1$). A carga de rede é dada por $\rho = \lambda/\mu = \lambda$ (Erlang), onde λ é a taxa de chegada do processo e μ a taxa de serviço [Tessinari, Colle e Garcia 2017, Mandelbaum e Zeltyn 2004]. Para cada par definido para conexão, o *RMSA* deve identificar os caminhos de trabalho e de *backup* e quantos *slots* são necessários para transportar a requisição. O número necessário de *slots* nos enlaces pertencentes às rotas selecionadas é procurado, respeitando a contiguidade do espectro e restrições de continuidade. A Tabela 1 apresenta o número de *slots* de acordo com a taxa de bit, considerando Deslocamento de Fase em Quadratura e Dupla Polarização (*DP – QPSK*) [Gerstel et al. 2012] e 10 GHz de banda de guarda entre os canais. Se o *RMSA* puder adicionar os recursos necessários da requisição, então esta é aceita. Do contrário, é rejeitada [Tessinari, Colle e Garcia 2017].

Tabela 1 – Número de *slots* de frequência de 12,5 Ghz para a modulação DP-QPSK [Tessinari, Colle e Garcia 2017] e taxa de transmissão de bit.

Taxa de Bit	Número de Slots
40 Gb/s	3
100 Gb/s	4
400 Gb/s	7
1 Tb/s	16

Neste trabalho, um cenário de esquema de proteção de caminho dedicado 1:1 com falhas foi adicionado. No cenário, quando uma nova requisição é criada, ela só será alocada

se houver recursos tanto para o caminho de trabalho como para o caminho de *backup*. Porém, os recursos são adicionados apenas no caminho de trabalho, não reservando-os no caminho de *backup*. No momento que ocorrer uma falha, todas as conexões já alocadas (se houver) são verificadas. Então:

- Conexões que estão alocadas no caminho de trabalho e possuem o enlace com falha são trocadas para o caminho de *backup*. Os recursos que haviam no caminho com falha de enlace são liberados para melhorar a utilização dos recursos. Se não for possível alocar o caminho de *backup* devido à falta de recursos, a conexão é interrompida e quando a falha é reparada a conexão não será restabelecida;
- Conexões que estão alocadas no caminho de trabalho mas não utilizam o enlace com falha permanecem inalteradas;
- Cada nova requisição deve ser alocada em seu respectivo caminho de *backup* no caso de um caminho de trabalho estar quebrado no tempo de alocação.

Na Figura 10 é apresentada uma simulação de uma topologia no *ElasticO++*. Os módulos: “Controlador”, “Gerador de Requisição”, “Gerador de Falhas” e o “*theEye*” que tem o objetivo de controlar tudo no *software*, estão localizados à esquerda. O número de requisições já atendidas e total e o número de falhas já ocorridas, são apresentados acima dos módulos. Na topologia, em cada enlace é exibido o número de *slots* disponíveis e o número de *slots* total, juntamente com a porcentagem atual de *slots* disponíveis. Estes enlaces mudam de cores a medida que os *slots* disponíveis vão diminuindo. Na janela inferior, é identificado o *log* de cada módulo em ordem de chamada. Neste exemplo, um processo de alocação aceito com sucesso é mostrado.

3.1.3 Contribuições na Ferramenta *ElasticO++*

A seguir são resumidas as implementações deste trabalho na ferramenta *ElasticO++*:

- Inserção do algoritmo *Suurballe e Tarjan*;
- Implementação dos algoritmos propostos: *Suurballe e Tarjan Mais Balanceado* e *Suurballe e Tarjan Menos Balanceado*;
- Inserção do novo módulo na arquitetura do *ElasticO++* chamado de “Gerador de Falhas”; e
- Inserção da técnica de proteção de caminho dedicado 1:1.

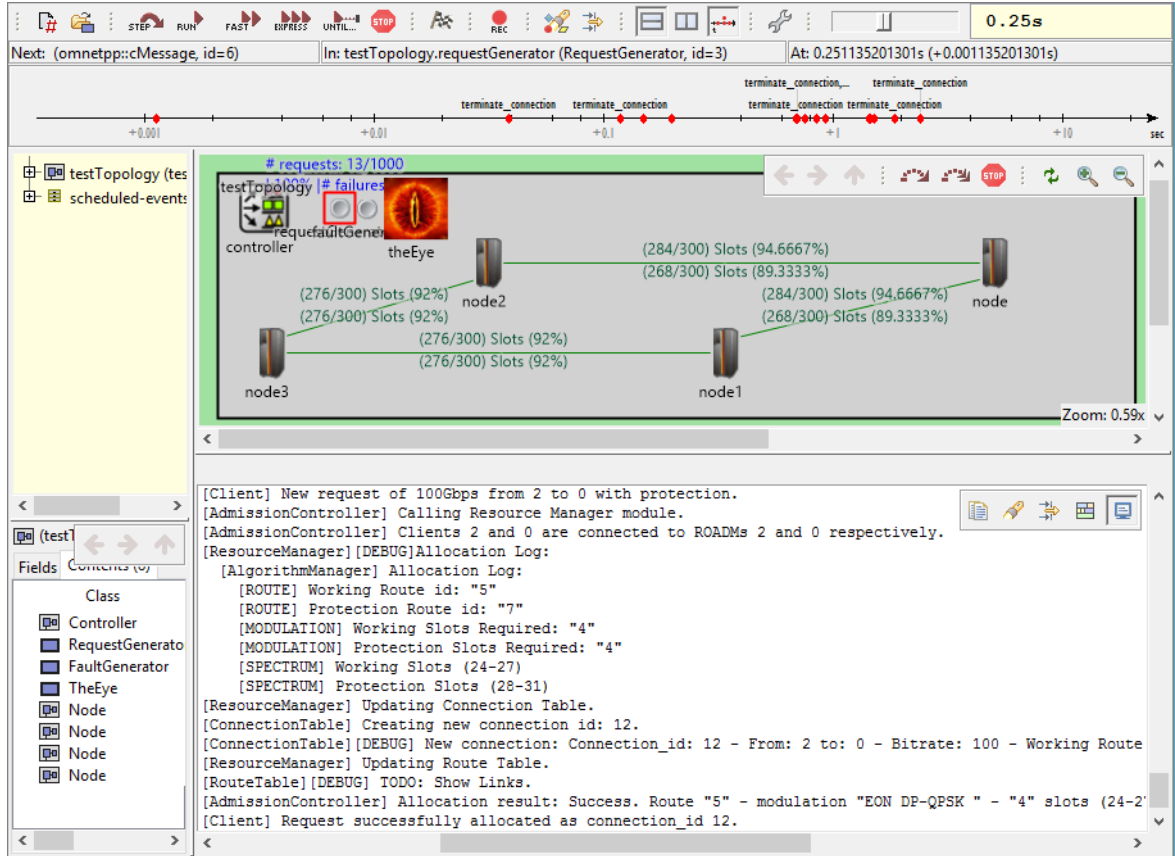


Figura 10 – Exemplo gráfico de uma simulação de uma topologia no *software ElasticO++*.

3.2 Algoritmo de Suurballe e Tarjan

Seja $G = (V, A)$ um grafo *2-aresta-conexo* de ordem $N = |V|$ onde $V = \{v_1, v_2, \dots, v_N\}$ é o conjunto de vértices, L é o número de arestas com $L = |A|$, onde A é o conjunto de arestas de pesos unitários. Para cada par de nodo origem, v_o , e nodo destino, v_d , de G , o algoritmo *ST* encontra o menor par de caminhos disjuntos por arestas para ser usado como caminhos de trabalho e de *backup* [Suurballe e Tarjan 1984]. De acordo com a versão de Whitney do Teorema de Menger [Whitney 1932], é possível encontrar dois caminhos sempre que a rede for *2-aresta-conexo* (ver Seção 1.1). A complexidade do tempo de execução do algoritmo *ST* é $O(L + N \log N)$. O algoritmo *ST* [Oliveira 2010] é descrito na sequência:

1. Executa o algoritmo de *Dijkstra* padrão [Dijkstra 1959], porém com uma modificação no critério de parada, que será somente quando o custo do nodo origem, v_o , para todos os outros nodos do grafo for calculado. Em seguida, define o menor caminho P_1 conectando o nodo origem, v_o , e o nodo destino, v_d ;
2. Mudam-se os pesos das arestas de P_1 . Este passo é conhecido como “transformação

de custos” e ocorre da seguinte forma:

$$c_{v_i, v_j} = \begin{cases} N & , se (v_i, v_j) \in P_1 \\ c_{v_i, v_j} + c_{v_i} - c_{v_j} & , se (v_i, v_j) \notin P_1 \\ & e (v_j, v_i) \notin P_1 \\ 0 & , se (v_j, v_i) \in P_1, \end{cases}$$

onde c_{v_i} é o custo do caminho do nodo v_o para o nodo v_i e c_{v_i, v_j} é o custo da aresta (v_i, v_j) . As arestas são bidirecionais, portanto temos uma aresta no sentido de v_i para v_j e uma aresta no sentido de v_j para v_i . Estas arestas terão pesos diferentes para a definição de um novo caminho. Quando uma aresta faz parte de P_1 , a aresta no sentido em que foi utilizada passa a ter custo infinito, atribuído com N neste trabalho. O sentido inverso dessa aresta, recebe custo zero. As arestas que não fazem parte de P_1 sofrem a transformação indicada. Por fim, o conjunto resultante contém as arestas das quais podem ser obtidos os dois menores caminhos disjuntos por arestas.

3. Executa o algoritmo de *Dijkstra* mais uma vez com os pesos das arestas de P_1 modificados e encontra o segundo menor caminho P_2 ;
4. Verifica se P_1 e P_2 compartilham alguma aresta. Se sim, P_1 e P_2 são redefinidos para P'_1 e P'_2 , onde P'_1 será formado pelas arestas iniciais de P_1 que antecedem a aresta compartilhada, seguido do final das arestas de P_2 que sucedem a aresta compartilhada. A aresta em comum não é adicionada. Para P_2 o método é o mesmo, iniciando pelo P_2 e finalizando com as arestas de P_1 . Este procedimento será repetido se nos caminhos houver mais de uma aresta em comum. Se não houver nenhuma aresta compartilhada em P_1 e P_2 , então eles permanecem.

O par de caminhos resultante, pode ser P_1 e P_2 ou P'_1 e P'_2 , é utilizado para definir os caminhos de trabalho e de *backup*.

Como exemplo, considere a rede apresentada na Figura 11(a), com $v_o = v_1$ e $v_d = v_8$ e arestas com pesos unitários. A distância de v_1 para qualquer outro nodo da rede será calculada. Com isso, o menor caminho P_1 é obtido: $v_1 \leftrightarrow v_5 \leftrightarrow v_4 \leftrightarrow v_8$ com três saltos (Figura 11(b)). Os pesos de todas as arestas são modificados seguindo a “transformação de custos” e estão apresentados na Figura 11(c). Um segundo menor caminho P_2 é encontrado: $v_1 \leftrightarrow v_2 \leftrightarrow v_3 \leftrightarrow v_4 \leftrightarrow v_5 \leftrightarrow v_6 \leftrightarrow v_7 \leftrightarrow v_8$ (Figura 11(d)). Em seguida, é verificado se P_1 e P_2 compartilham arestas, neste exemplo, $v_4 \leftrightarrow v_5$. Portanto, P_1 e P_2 são redefinidos como P'_1 e P'_2 , respectivamente. Na Figura 11 (e) são ilustrados ambos os caminhos finais: P'_1 com quatro saltos $v_1 \leftrightarrow v_5 \leftrightarrow v_6 \leftrightarrow v_7 \leftrightarrow v_8$ e P'_2 também com quatro saltos $v_1 \leftrightarrow v_2 \leftrightarrow v_3 \leftrightarrow v_4 \leftrightarrow v_8$.

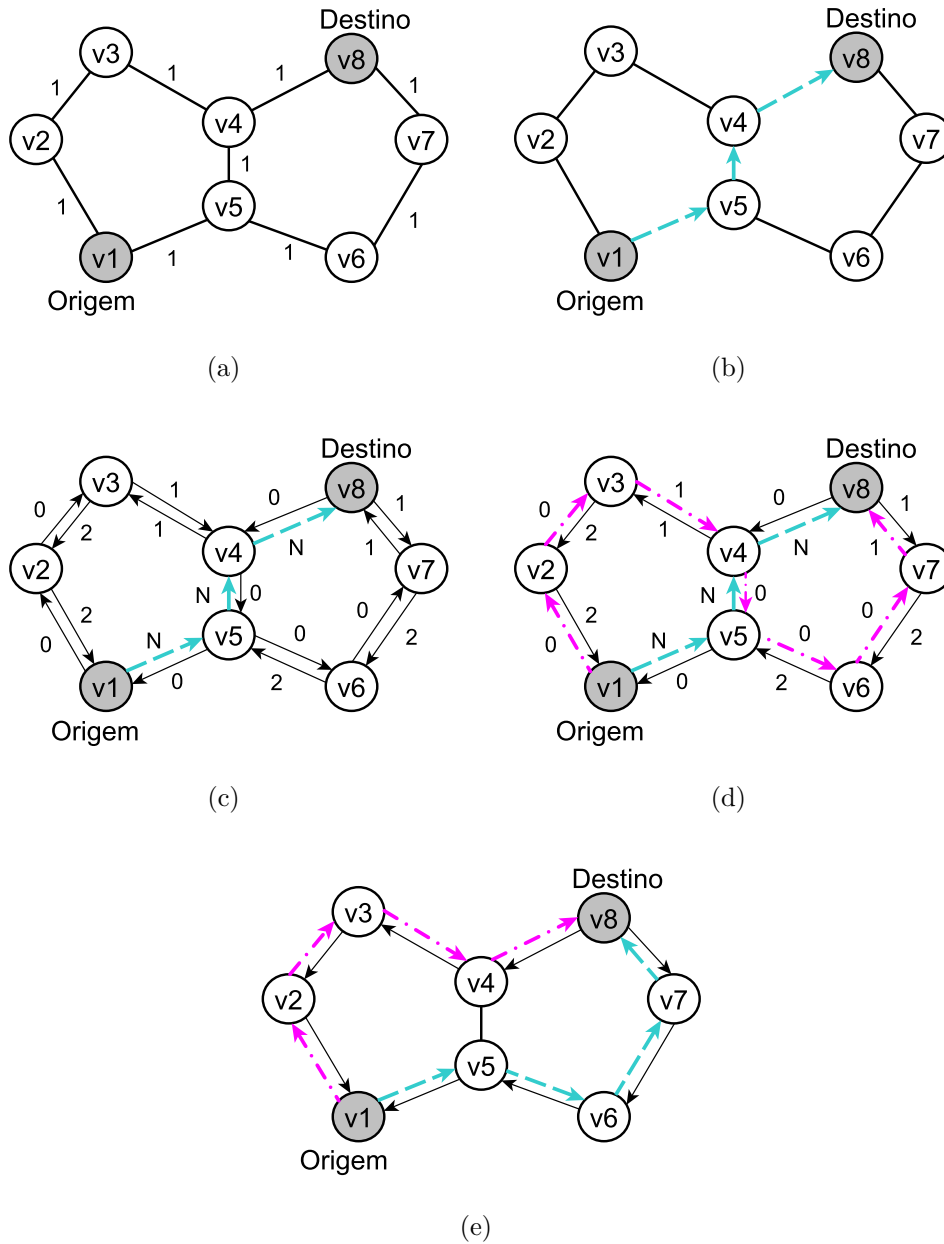


Figura 11 – (a) Topologia de rede; De $v_o = v_1$ para $v_d = v_8$: (b) menor caminho (em linha tracejada azul) obtido com o algoritmo de *Dijkstra*, (c) atualização dos pesos das arestas de acordo com o passo da “transformação de custos”, (d) segundo menor caminho (em linha com ponto e traço rosa), obtidos com o algoritmo de *Dijkstra*, (e) caminhos disjuntos de trabalho e de *backup* obtidos com o algoritmo *ST*.

3.3 Algoritmo de Yen

O algoritmo de *Yen* calcula K menores caminhos entre um par de nodos origem e destino. Em cada caminho, os nodos não podem ser repetidos. Para a definição de K caminhos, o menor caminho possível é obtido e a partir dele, novos menores caminhos são descobertos. Subsequentemente a partir do segundo menor caminho ocorre a busca de

novos menores caminhos e assim sucessivamente até K ou até definir todos os menores caminhos possíveis. A implementação geralmente utilizada segue a versão clássica, a qual busca pelo menor caminho entre um par de nodos origem e destino e em seguida, remove as arestas do menor caminho uma por vez. Cada vez que uma aresta é removida, um novo caminho é encontrado. A aresta removida é adicionada novamente antes da remoção da próxima aresta. Quando o algoritmo finalizar a última aresta do menor caminho, ele inicia o processo novamente pelo segundo menor caminho, e continua até encontrar K caminhos suficientes [Yen 1970].

Neste trabalho, foi utilizada a abordagem de [Martins e Pascoal 2003], onde os nodos de cada caminho são analisados por ordem inversa. Esta abordagem é apresentada a seguir.

Sendo um grafo $G = (V, A)$ direcionado de ordem $N = |V|$ onde $V = \{v_1, v_2, \dots, v_N\}$ é o conjunto de vértices e o número de arestas L , com $L = |A|$ que é o conjunto de arestas e assumindo que não existam ciclos no grafo com custos negativos, então: v_o é o nodo de origem e v_d é o nodo destino; P_i é um caminho; $X = \{P_1, P_2, \dots, P_K\}$ é o conjunto de caminhos candidatos que existem do nodo v_o ao nodo v_d para identificar novos caminhos; a árvore é um sub-caminho de um futuro caminho de X ; o custo da aresta v_i a v_j será c_{v_i, v_j} e o custo de um nodo v_i será π_{v_i} , chamado de rótulo do nodo v_i .

Dado um par de nodos de uma topologia de rede, o algoritmo de Yen segue estes passos:

1. Encontra o menor caminho P_i através do algoritmo de *Dijkstra*. Este caminho não estará no conjunto X .
2. Remove todos os nodos que pertençam a P_i com exceção do nodo v_d ;
3. Encontra uma árvore com o menor custo a partir da rede que ficou;
4. Adiciona o nodo antecessor (v_i) a v_d do caminho P_i na rede e suas arestas. Se ele for v_o , não haverá mais nodos para adicionar. A aresta que conecta v_i e v_d não deverá ser adicionada nesse momento. Quando um nodo é re-inserido, o rótulo desse nodo é atualizado somente se $\pi_{v_i} > \pi_{v_j} + c_{v_i, v_j}$. Se sim, $\pi_{v_i} = \pi_{v_j} + c_{v_i, v_j}$ e os sucessores desse nodo também são atualizados se $\pi_{v_j} > \pi_{v_i} + c_{v_j, v_i}$, para $\pi_{v_j} = \pi_{v_i} + c_{v_j, v_i}$;
5. Encontra uma árvore com o menor custo a partir da rede que ficou, e com ela um novo menor caminho P_{i+1} será definido. Este caminho será adicionado ao conjunto X e é chamado de candidato, pois poderá definir novos caminhos no algoritmo mais a frente;
6. Re-adiciona a aresta (v_i, v_d) . Se $\pi_{v_i} > \pi_{v_d} + c_{v_i, v_d}$ então $\pi_{v_i} = \pi_{v_d} + c_{v_i, v_d}$ e atualiza os rótulos dos sucessores se $\pi_{v_d} > \pi_{v_i} + c_{v_d, v_i}$, para $\pi_{v_d} = \pi_{v_i} + c_{v_d, v_i}$;

7. Prossegue com os Itens 4, 5 e 6 enquanto houver nodos do caminho de P_i para adicionar $(v_{i-1}, v_{i-2}, \dots, v_o)$;
8. Re-insere todas as arestas novamente quando o processo de adicionar nodos acabar;
9. Adiciona em X todos os “ $P_{i's}$ ” encontrados. Após a rede ser restaurada, o P_i com menor custo sairá de X e novos caminhos serão encontrados a partir dele, retornando ao Item 2;
10. O algoritmo termina quando este atingir o número K de caminhos ou todos os caminhos possíveis já forem definidos.

Este algoritmo tem a mesma complexidade do algoritmo de *Yen* clássico $O(KN(L + N \log N))$. A diferença é que esta nova versão tem um melhor desempenho em casos médios. O pseudocódigo e mais informações sobre esta versão podem ser encontrados em [Martins e Pascoal 2003].

Como exemplo, considere a rede da Figura 12(a).

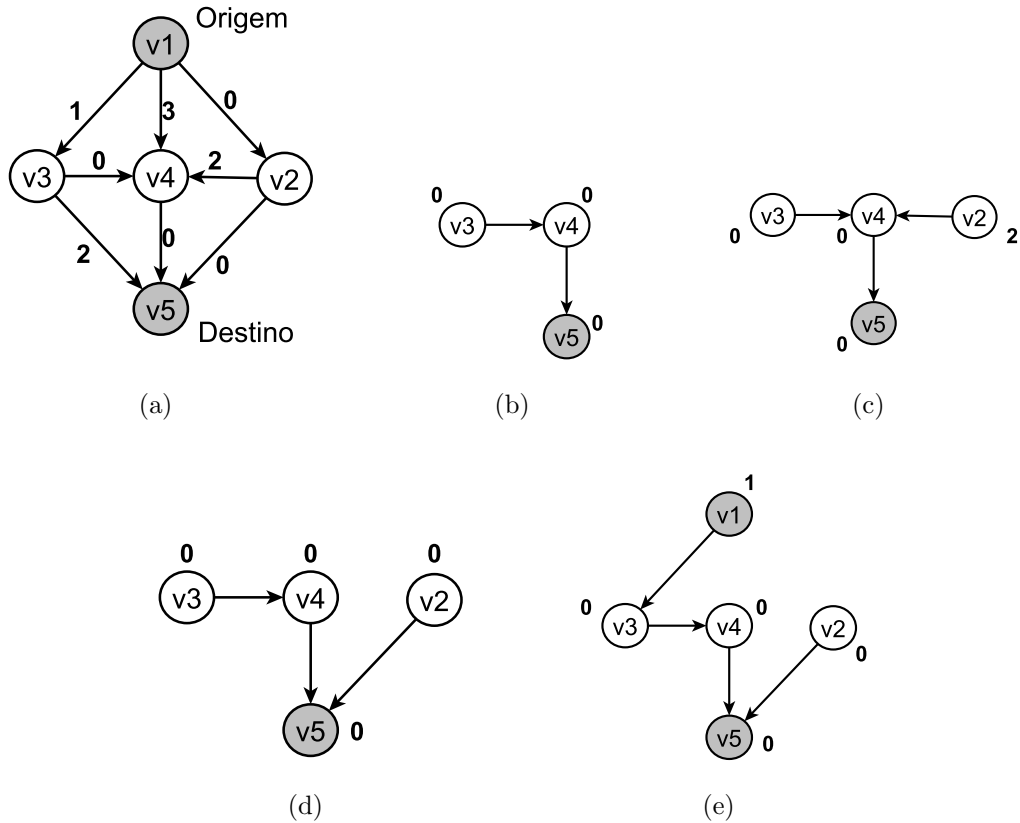


Figura 12 – Processo de busca de K menores caminhos do par de nodos $v_o = v_1, v_d = v_5$. Adaptado de [Martins e Pascoal 2003].

Supondo que o menor caminho definido pelo algoritmo de *Dijkstra* seja $P_1 = v_1 \leftrightarrow v_2 \leftrightarrow v_5$. Em seguida, todos os nodos e arestas de P_1 com exceção do nodo final v_5 , são

removidos da topologia. Então a árvore em v_5 com menor custo é definida. Esta parte está apresentada na Figura 12(b). Em seguida o nodo v_2 é adicionado na rede, o rótulo do nodo v_2 é atualizado ($\pi_2 = \pi_4 + c_{2,4}$) e os nodos sucessores também, o resultado é apresentado na Figura 12(c). Uma vez que v_2 é rotulado, a árvore com menor custo é calculada e um novo menor caminho é encontrado: $P_2 = v_1 \leftrightarrow v_2 \leftrightarrow v_4 \leftrightarrow v_5$ e armazenado em X com custo total de caminho igual a 2.

A aresta (v_2, v_5) deve ser re-inserida na topologia. Isso implica em atualizar o rótulo do nodo v_2 e possivelmente os nodos sucessores. Após a atualização a árvore com menor custo está representada na Figura 12(d). Em seguida o nodo v_1 é restaurado e a árvore com menor custo obtida é a da Figura 12(e), encontrando o caminho $P_3 = v_1 \leftrightarrow v_3 \leftrightarrow v_4 \leftrightarrow v_5$ com custo total de caminho igual a 1. Este caminho é armazenado em X . A sequência do algoritmo se dá a partir de P_3 que possui o menor custo de caminho e sai do conjunto X retornando ao processo de busca inicial de menores caminhos. Este processo continua até que todos os K menores caminhos requisitados sejam encontrados.

3.4 Conclusão

Foi apresentada aqui, a ferramenta *ElasticO++* que habilita testes, simulações e comparações de algoritmos em cenários de redes ópticas elásticas. A sua arquitetura e seus componentes principais foram destacados e também, as contribuições realizadas a partir deste trabalho na ferramenta. Além disso, foram descritos o passo a passo de dois algoritmos que buscam menores caminhos a partir de um par de nodos origem e destino, o algoritmo *Suurballe e Tarjan (ST)*, implementado a partir deste trabalho, e o algoritmo do *Yen*. Em adição, exemplificações de seus funcionamentos foram mostradas.

4 Algoritmos Propostos

Este capítulo apresenta os dois algoritmos propostos no contexto de redes ópticas elásticas. Estes algoritmos são versões do algoritmo *Suurballe e Tarjan (ST)* e foram desenvolvidos para aproveitar a diversidade de comprimentos de caminhos de trabalho e de *backup* interconectando o mesmo par de nodos origem e destino em uma topologia.

As versões propostas são chamadas de Suurballe e Tarjan Mais Balanceado (*STMB*) e Suurballe e Tarjan Menos Balanceado (*STLB*) e têm o objetivo de definir comprimentos de caminhos mais balanceados e menos balanceados. Estes algoritmos serão úteis para projetistas e pesquisadores identificarem diferenças que podem existir quando não há essa opção de escolha de comprimentos de caminhos com o *Suurballe e Tarjan*. O fluxograma e o passo a passo dos algoritmos são apresentados a seguir.

4.1 Suurballe e Tarjan Mais Balanceado

O objetivo do algoritmo *STMB* é encontrar o menor par de caminhos disjuntos por arestas para cada par de nodos origem e destino, mantendo os comprimentos dos caminhos de trabalho e de *backup* com a menor diferença possível. Os comprimentos destes caminhos serão denominados a partir de agora de W^+ (comprimento do caminho de trabalho mais balanceado) e B^+ (comprimento do caminho de *backup* mais balanceado). Além disso, a soma destes comprimentos será denominada C .

O fluxograma do algoritmo é apresentado na Figura 13 e segue desta forma:

1. Executa o algoritmo *ST* padrão para uma rede *2-aresta-conexa*. Para cada par origem e destino, dois caminhos disjuntos por arestas são encontrados. Então C , que é a soma dos comprimentos destes caminhos, será armazenado;
2. Define as variáveis auxiliares R e S de cada par baseado em C , onde R definirá o novo comprimento de W^+ e seu valor será o arredondamento para baixo da metade de C . O S definirá o novo comprimento de B^+ e seu valor será o resultado de C menos R . Esta etapa sempre será possível porque C é um número inteiro positivo medido em número de saltos;
3. Cria uma lista L para armazenar todos os caminhos de comprimento R para um dado par de nodos origem e destino, usando o algoritmo de Yen [Yen 1970]. Para isso, os caminhos em ordem crescente de tamanho são apresentados e quando surgir

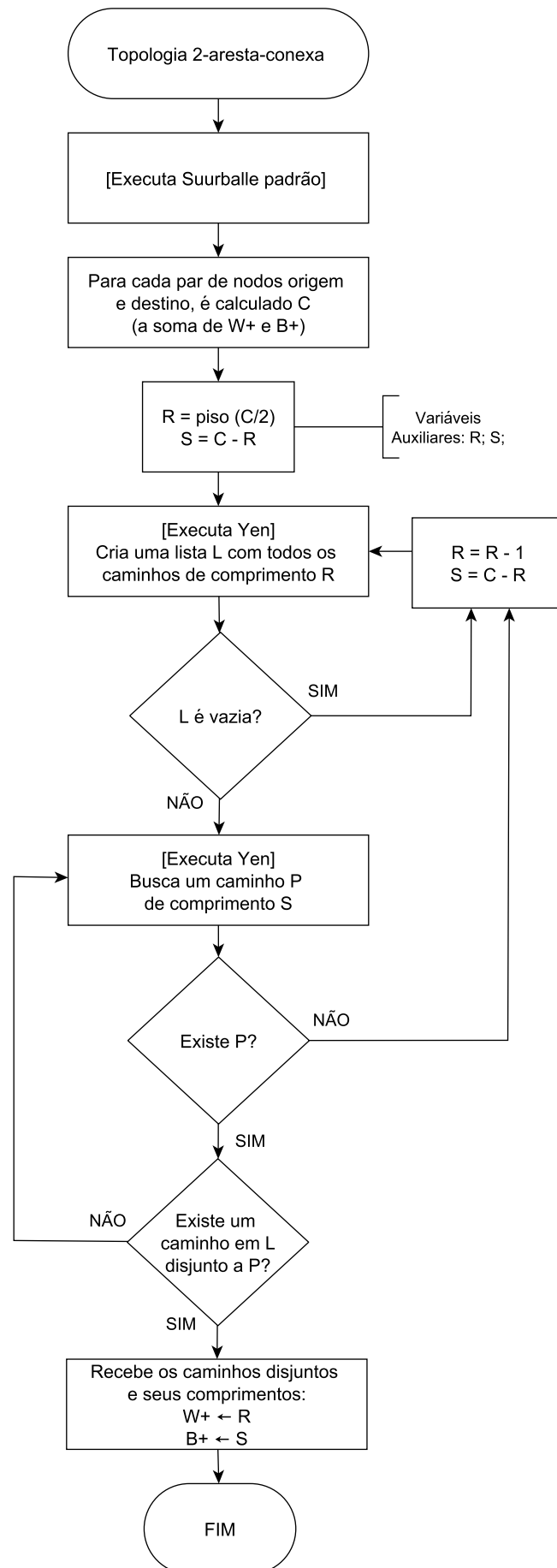


Figura 13 – Fluxograma do algoritmo Suurballe e Tarjan Mais Balanceado (STMB).

o primeiro caminho de comprimento R , a lista deixará de ser vazia e armazenará o caminho. Os próximos caminhos de comprimento R serão adicionados também. Aqui, o algoritmo de *Yen* não possui um número K específico como critério de parada. Então, é apresentado um caminho por vez, até que o caminho apresentado possua um comprimento maior que R ou todos os caminhos possíveis para determinado par origem e destino terminarem. Por ser um algoritmo exaustivo, o algoritmo de *Yen* sempre será viável;

4. Verifica se L está vazia. Se estiver, nenhum caminho de comprimento R foi encontrado e então o R será decrementado e o algoritmo retorna ao Item 3. Se não estiver, segue para o Item 5;
5. Busca um e somente um caminho P de comprimento S , também usando o algoritmo de *Yen*.
6. Verifica se existe um caminho P de comprimento S . Se não existir, o R é decrementado e retorna ao Item 3. Se existir, segue para o Item 7;
7. Compara o caminho P de comprimento S com o primeiro caminho de L . Nesse momento, é verificado se existe alguma aresta compartilhada entre estes caminhos. Se não existir aresta compartilhada, os caminhos são recebidos e os valores de R e S são atribuídos a W^+ e B^+ , respectivamente. Se existir uma aresta compartilhada, o próximo caminho de L é testado. Se nenhum caminho de L for disjunto por arestas ao caminho P de comprimento S , então retorna ao Item 5 para buscar outro caminho de comprimento S . Se todos os caminhos de comprimento S forem testados e nenhum for disjunto por arestas a algum caminho de L , então R é decrementado e o algoritmo retorna ao Item 3. Esse procedimento será realizado até que dois caminhos disjuntos por arestas sejam encontrados.

Se houver dois caminhos disjuntos por arestas em cada par de nodos origem e destino da topologia, o algoritmo *ST* será bem sucedido e consequentemente o *STMB* também será.

Como exemplo, considere a topologia de rede *2-aresta-conexa* apresentada na Figura 14. O algoritmo *ST* é chamado e todos os comprimentos de caminhos de trabalho e de *backup* são obtidos para todos os pares de nodos origem e destino. Estes comprimentos são somados e o C de cada par origem e destino é definido. Considere o par com origem v_8 e destino v_{17} . Para este par, o comprimento mínimo é $C = 16$.

Utilizando o *STMB*, o R será $16/2 = 8$ e o S será $16 - 8 = 8$. Uma lista L com todos os caminhos de comprimento R é criada. A lista não será vazia, pois é possível encontrar caminhos de comprimento 8, como: $L_8(a): v_8 \leftrightarrow v_7 \leftrightarrow v_6 \leftrightarrow v_5 \leftrightarrow v_4 \leftrightarrow v_3 \leftrightarrow v_2 \leftrightarrow v_1 \leftrightarrow v_{17}$; $L_8(b): v_8 \leftrightarrow v_9 \leftrightarrow v_{10} \leftrightarrow v_{11} \leftrightarrow v_{12} \leftrightarrow v_{13} \leftrightarrow v_7 \leftrightarrow v_{16} \leftrightarrow v_{17}$; e

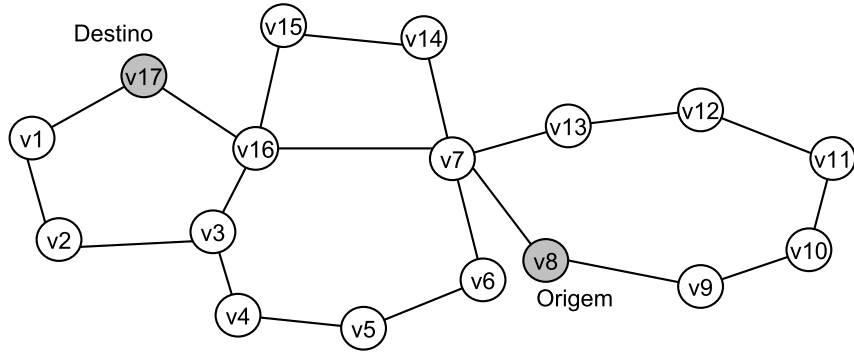
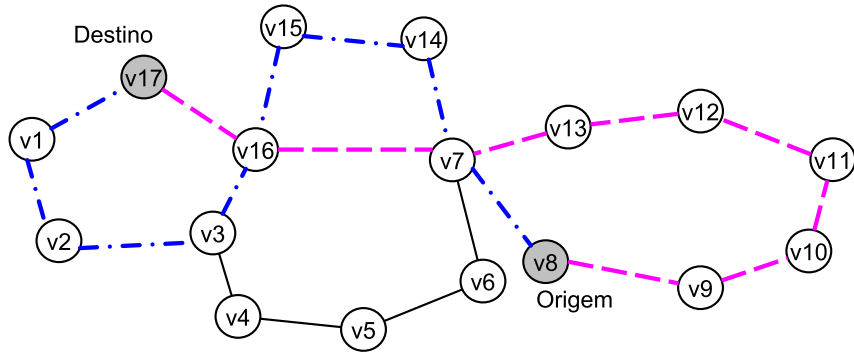


Figura 14 – Topologia 2-aresta-conexa exemplo.

$L_8(c)$: $v_8 \leftrightarrow v_7 \leftrightarrow v_{14} \leftrightarrow v_{15} \leftrightarrow v_{16} \leftrightarrow v_3 \leftrightarrow v_2 \leftrightarrow v_1 \leftrightarrow v_{17}$. Na sequência, é buscado um caminho P de comprimento S disjunto por arestas ao primeiro caminho da lista L . É verificada a disjunção. Considerando o par origem v_8 e destino v_{17} , o determinado caminho de comprimento R : $L_8(a)$ e o determinado caminho de comprimento S : P_1 : $v_8 \leftrightarrow v_7 \leftrightarrow v_{14} \leftrightarrow v_{15} \leftrightarrow v_{16} \leftrightarrow v_3 \leftrightarrow v_2 \leftrightarrow v_1 \leftrightarrow v_{17}$ não são disjuntos por arestas. Então segue para o próximo caminho de L . Sendo o caminho de comprimento R : $L_8(b)$ e o caminho de comprimento S : P_1 (Figura 15). Estes caminhos são disjuntos por arestas, portanto os caminhos são recebidos, W^+ recebe o R , B^+ recebe o S e o algoritmo acaba.

Figura 15 – Dois caminhos disjuntos por arestas com comprimentos $R = 8$ (em linha tracejada rosa) e $S = 8$ (em linha com ponto e traço azul) do par origem v_8 para o destino v_{17} .

Se acaso não houvesse nenhum caminho em L de comprimento $R = 8$ disjunto por arestas a um caminho P de comprimento $S = 8$, outro caminho P de mesmo comprimento S seria buscado. Se acaso nenhum caminho P fosse disjunto por arestas a algum caminho em L , haveria o decremento de R e atualização de S . Portanto, seria $R = 7$ e $S = 9$. Uma nova lista L com todos os caminhos com comprimentos de tamanho R seria criada, e caminhos de comprimento R seriam encontrados, como: $L_7(a)$: $v_8 \leftrightarrow v_7 \leftrightarrow v_6 \leftrightarrow v_5 \leftrightarrow v_4 \leftrightarrow v_3 \leftrightarrow v_{16} \leftrightarrow v_{17}$. Um caminho de comprimento S seria buscado. Para este caso não seria encontrado nenhum caminho com comprimento S . Então R é decrementado e S atualizado mais uma

vez. Então, seria $R = 6$ e $S = 10$. Uma lista L com caminhos de comprimento R seria criada, sendo: $L_6(a)$: $v_8 \leftrightarrow v_7 \leftrightarrow v_{16} \leftrightarrow v_3 \leftrightarrow v_2 \leftrightarrow v_1 \leftrightarrow v_{17}$. Um caminho de comprimento S seria encontrado, sendo: P_1 : $v_8 \leftrightarrow v_9 \leftrightarrow v_{10} \leftrightarrow v_{11} \leftrightarrow v_{12} \leftrightarrow v_{13} \leftrightarrow v_7 \leftrightarrow v_{14} \leftrightarrow v_{15} \leftrightarrow v_{16} \leftrightarrow v_{17}$. Os caminhos de comprimento R : $L_6(a)$ e de comprimento S : P_1 são disjuntos por arestas (Figura 16). Os caminhos são recebidos, W^+ recebe o R , B^+ recebe o S e o algoritmo acaba.

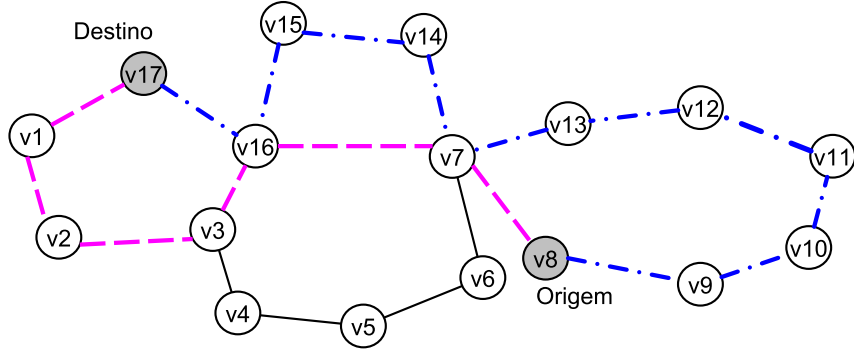


Figura 16 – Dois caminhos disjuntos por arestas com comprimentos $R = 6$ (em linha tracejada rosa) e $S = 10$ (em linha com ponto e traço azul) do par origem v_8 para o destino v_{17} .

4.2 Suurballe e Tarjan Menos Balanceado

Diferentemente do *STMB*, o objetivo do algoritmo *STLB* é encontrar o menor par de caminhos disjuntos por arestas para cada par de nodos origem e destino, mantendo os comprimentos dos caminhos de trabalho e de *backup* com a maior diferença possível. Os comprimentos destes caminhos serão denominados a partir de agora de W^- (comprimento do caminho de trabalho menos balanceado) e B^- (comprimento do caminho de *backup* menos balanceado).

O algoritmo *STLB* age análogo ao *STMB* mas com duas diferenças: *i*) para obter o menor caminho de trabalho possível, R inicia com o comprimento do menor caminho encontrado pelo algoritmo de *Dijkstra* [Dijkstra 1959]; e *ii*) quando dois caminhos disjuntos por arestas com comprimentos R e S não são encontrados, R é incrementado. Como no *STMB*, se houver dois caminhos disjuntos por arestas para cada par de nodos origem e destino na topologia, o algoritmo *ST* será bem sucedido e consequentemente o *STLB* também será.

Como exemplo, considere a topologia de rede *2-aresta-conexa* apresentada na Figura 14. O algoritmo *ST* é chamado e todos os comprimentos de caminhos de trabalho e de *backup* são obtidos para todos os pares de nodos origem e destino. Estes comprimentos são somados e o C de cada par origem e destino é definido. Considere o par com origem v_8 e destino v_{17} . Para este par, o comprimento mínimo é $C = 16$.

Utilizando o *STLB*, o R é definido pelo algoritmo de *Dijkstra* e será 3 e o S será $16 - 3 = 13$. Uma lista L com todos os caminhos de comprimento R é criada. A lista não será vazia, pois é possível encontrar caminhos de comprimento 3, como: $L_3(a)$: $v_8 \leftrightarrow v_7 \leftrightarrow v_{16} \leftrightarrow v_{17}$. Na sequência, é buscado um caminho P de comprimento S disjunto por arestas ao primeiro caminho da lista L , sendo: P_1 : $v_8 \leftrightarrow v_9 \leftrightarrow v_{10} \leftrightarrow v_{11} \leftrightarrow v_{12} \leftrightarrow v_{13} \leftrightarrow v_7 \leftrightarrow v_{14} \leftrightarrow v_{15} \leftrightarrow v_{16} \leftrightarrow v_3 \leftrightarrow v_2 \leftrightarrow v_1 \leftrightarrow v_{17}$. É verificada a disjunção. Considerando o par origem v_8 e destino v_{17} , o determinado caminho de comprimento R : $L_3(a)$ e o determinado caminho de comprimento S : P_1 (Figura 17) são disjuntos por arestas, portanto os caminhos são recebidos, W^- recebe o R , B^- recebe o S e o algoritmo acaba.

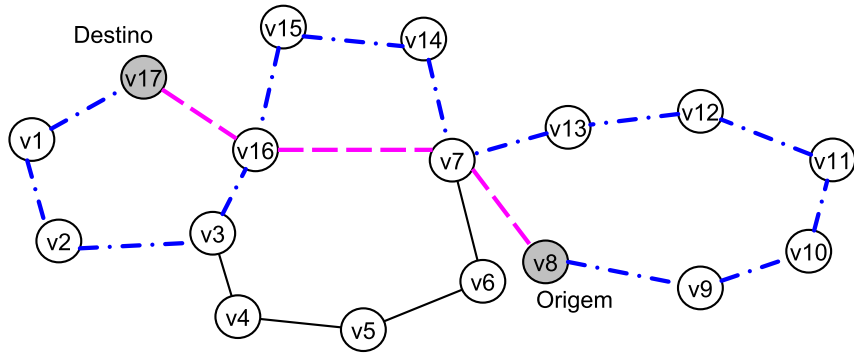


Figura 17 – Dois caminhos disjuntos por arestas com comprimentos $R = 3$ (em linha tracejada rosa) e $S = 13$ (em linha com ponto e traço azul) do par origem v_8 para o destino v_{17} .

Se acaso não houvesse nenhum caminho em L de comprimento $R = 3$ disjunto por arestas a um caminho P de comprimento $S = 13$, outro caminho P de mesmo comprimento S seria buscado. Se acaso nenhum caminho P fosse disjunto por arestas a algum caminho em L , haveria o incremento de R e atualização de S . Portanto seria $R = 4$ e $S = 12$. Uma nova lista L com todos os caminhos com comprimentos de tamanho R seria criada se $R = 4$ existisse. Como não é o caso, L é vazia e portanto, o R seria incrementado e o S atualizado. Com $R = 5$ e $S = 11$, uma nova lista L com todos os caminhos com comprimentos de tamanho R seria criada, e caminhos de comprimento R seriam encontrados, como: $L_5(a)$: $v_8 \leftrightarrow v_7 \leftrightarrow v_{14} \leftrightarrow v_{15} \leftrightarrow v_{16} \leftrightarrow v_{17}$. Um caminho de comprimento S seria buscado e encontrado, como: P_1 : $v_8 \leftrightarrow v_9 \leftrightarrow v_{10} \leftrightarrow v_{11} \leftrightarrow v_{12} \leftrightarrow v_{13} \leftrightarrow v_7 \leftrightarrow v_{16} \leftrightarrow v_3 \leftrightarrow v_2 \leftrightarrow v_1 \leftrightarrow v_{17}$. Os caminhos de comprimento R : $L_5(a)$ e de comprimento S : P_1 são disjuntos por arestas (Figura 18). Os caminhos são recebidos, W^- recebe o R , B^- recebe o S e o algoritmo acaba.

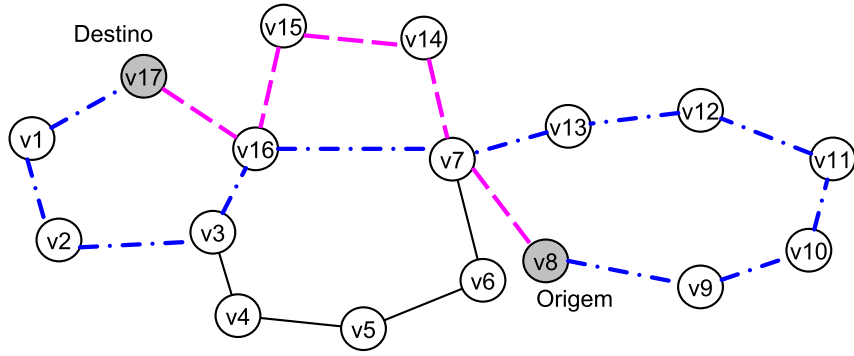


Figura 18 – Dois caminhos disjuntos por arestas com comprimentos $R = 5$ (em linha tracejada rosa) e $S = 11$ (em linha com ponto e traço azul) do par origem v_8 para o destino v_{17} .

4.3 Tempo Computacional

Dado um par de nodos origem e destino, os algoritmos *STMB* e *STLB* não necessariamente buscarão todos os pares de caminhos possíveis de comprimento C . Assim que encontrarem os caminhos disjuntos por arestas de comprimentos R e S , mesmo se esses forem modificados por incremento ou decremento, os algoritmos terminam. Estes caminhos podem ser os mesmos definidos pelo *ST* ou não.

Se os algoritmos definem determinados comprimentos R e S que existem para o par origem e destino e algum caminho de comprimento R seja disjunto por arestas a algum caminho de comprimento S , o algoritmo executa rapidamente, sendo este o melhor caso. Já se os algoritmos definem determinados comprimentos R e S que não existem ou que os caminhos de comprimento R e S não são disjuntos por arestas, muitas verificações poderão serão feitas, principalmente se houver a necessidade de incrementar ou decrementar o R e o S várias vezes. Se os novos W e B forem encontrados nas últimas opções possíveis de comprimento de caminhos, é considerado como o pior caso.

Por exemplo, considerando que todos os caminhos de comprimento R que fazem parte de uma lista L_R foram testados para todos os possíveis caminhos P de comprimento S (sendo L_S a lista de caminhos encontrados de comprimento S), e que o R e o S serão atualizados n_{att} vezes, então $n_{att} * |L_R| * |L_S|$ é o número de verificações no pior caso. Assim, os caminhos disjuntos por arestas serão o último caminho de L_R e o último P encontrado em L_S .

Os algoritmos *STMB* e *STLB* utilizam o algoritmo *Suurballe e Tarjan* padrão, em seguida buscam um caminho com um determinado comprimento e depois buscam outro caminho com outro determinado comprimento. A complexidade do algoritmo *Suurballe e Tarjan* é $O(L + N \log N)$ e a complexidade de buscar caminhos até encontrar um com um comprimento específico é $O(2^K)$ [Williams 2009]. Este problema foi provado

ser NP-Completo [Nykänen e Ukkonen 2002]. A complexidade de buscar caminhos até encontrar um com um comprimento específico é maior e portanto, é definida como a complexidade para os algoritmos *STMB* e *STLB*.

4.4 Vantagens e Desvantagens

Os diferentes comprimentos de caminhos de trabalho e de *backup* encontrados no algoritmo de *ST* em um mesmo par de nodos origem e destino, apresentam algumas vantagens e desvantagens.

Quando os caminhos possuem comprimentos com a menor diferença possível, são obtidos vantagens no momento de falhas, pois o tráfego trocado para o caminho de *backup*, não sofre com um caminho mais longo. Mas enquanto não houver falhas e houver diversidade de comprimentos de caminhos, é obtido a desvantagem de utilizar um caminho de trabalho que não seja o mais curto possível. Já, quando os caminhos possuem comprimentos com a maior diferença possível, a vantagem está em utilizar o caminho mais curto para o caminho de trabalho e consequentemente, a desvantagem está em utilizar o caminho mais longo após uma falha.

4.5 Conclusão

Foram apresentados duas versões do algoritmo *Suurballe e Tarjan*, uma que encontra os comprimentos do caminho de trabalho e de *backup* com a menor diferença possível, chamada de Suurballe e Tarjan Mais Balanceado e outra que encontra os comprimentos do caminhos de trabalho e de *backup* com a maior diferença possível, chamada de Suurballe e Tarjan Menos Balanceado. Ambos os algoritmos e a complexidade foram detalhados neste capítulo. Além disso, foi identificado que num cenário sem falhas, o *STLB* pode obter mais vantagens. Do contrário, o *STMB* pode ser mais vantajoso.

5 Metodologia dos Testes

Este capítulo aborda a metodologia de testes utilizada neste trabalho, apresentando cada passo realizado para que os resultados referentes as versões propostas do algoritmo *Suurballe e Tarjan (ST)* fossem atingidos.

5.1 Introdução

Para desenvolver os algoritmos de roteamento propostos: Suurballe e Tarjan Mais Balanceado e Suurballe (*STMB*) e Tarjan Menos Balanceado (*STLB*) foi utilizada a ferramenta *ElasticO++* [Tessinari et al. 2016]. Nesta ferramenta foi possível desenvolver e executar os algoritmos e obter os dados necessários para analisar o comportamento, impacto e diferenças do *STMB* e *STLB*. Além disso, o processo de alocação de recursos dado pelo *software* foi utilizado e introduzido num cenário de falhas com proteção de caminho dedicado 1:1.

Os testes relacionados aos algoritmos foram divididos em três etapas: *i*) em cada par de nodos origem e destino: obtenção de dados do comprimento dos caminhos de trabalho e de *backup* do *ST*, do *STMB* (W^+ e B^+) e do *STLB* (W^- e B^-); a diferença entre o W^+ com o W^- ; a diferença entre o B^- com o B^+ ; a diferença entre o B^- com o W^- ; a diferença entre o B^+ com o W^+ ; e o somatório de cada uma dessas diferenças; *ii*) verificação do impacto no número de *transponders*, e também no coeficiente de proteção; e *iii*) simulação de falhas em um cenário de proteção de caminho dedicado 1:1.

5.2 Procedimentos Metodológicos

A ferramenta *ElasticO++* desenvolvida a partir do simulador de eventos *OMNeT++* utiliza a linguagem *C++* e já possui alguns algoritmos implementados. Entre eles está o algoritmo de *Yen*. O algoritmo *ST* padrão e os algoritmos propostos *STMB* e *STLB* foram adicionados. Além disso, a alocação de recursos ocorreu de acordo com o esquema proteção de caminho dedicado 1:1, também adicionado no *ElasticO++*.

Os algoritmos propostos tem como entrada uma topologia de rede *2-aresta-conexa* e como saída fornecem os dois caminhos disjuntos por arestas e em um arquivo de extensão *CSV* (Valor Delimitado por Vírgula) os dados relativos aos comprimentos de caminhos de trabalho e de *backup* de cada par de nodos origem e destino obtidos com os algoritmos *ST* e *STMB* ou *ST* e *STLB* são armazenados.

5.2.1 Testes: Etapa 01

Os dados de saída dos arquivos de extensão *CSV* do *STMB* são: comprimentos de caminhos de trabalho e de *backup* do *ST* e comprimentos de caminhos de trabalho e de *backup* do *STMB* de cada par origem e destino da topologia de rede utilizada. Já no *STLB* são: comprimentos de caminhos de trabalho e de *backup* do *ST* e comprimentos de caminhos de trabalho e de *backup* do *STLB* de cada par origem e destino da topologia de rede utilizada. Com estes dados de cada rede, todas as diferenças entre os comprimentos de caminhos de trabalho e de *backup* de cada algoritmo, as diferenças entre os comprimentos dos caminhos dos algoritmos e o somatório dessas diferenças foram possíveis de obter. A seguir, estas métricas são apresentadas.

5.2.1.1 Métricas Para Identificar Diferenças Entre Comprimentos do STMB e STLB

Deve-se notar que para qualquer par de nodos, o W^- sempre será menor que ou igual ao W^+ , e o B^- sempre será maior que ou igual ao B^+ . Além disso, uma vez que ambos *STMB* e *STLB* definem pares de caminhos com o mesmo comprimento total C , se tem:

$$W^+ + B^+ = W^- + B^- = C. \quad (4)$$

A diferença entre as duas estratégias pode ser medida pela variação no comprimento dos caminhos de trabalho (ou caminhos de *backup*). Portanto, para cada par de nodos origem e destino (o, d):

$$\Delta_{o,d} = W^+ - W^- = B^- - B^+. \quad (5)$$

Em cada algoritmo, a diferença entre o comprimento do caminho de trabalho e o comprimento do caminho de *backup* para cada par de nodos (o, d) é dada por:

$$\begin{aligned} \Theta_{o,d}^+ &= B^+ - W^+; \\ \Theta_{o,d}^- &= B^- - W^-. \end{aligned} \quad (6)$$

Esta diferença entre comprimentos de caminho de trabalho e de *backup* é relacionada à diferença entre as duas estratégias. Esse relacionamento é dado por:

$$\Delta_{o,d} = \frac{\Theta_{o,d}^- - \Theta_{o,d}^+}{2}. \quad (7)$$

Portanto, quanto maior a diferença entre $\Theta_{o,d}^-$ e $\Theta_{o,d}^+$, maior a diferença entre os dois algoritmos.

A relação da diferença normalizada entre os comprimentos de caminhos de trabalho do *STMB* com comprimentos de caminhos de trabalho do *STLB*, é dada por:

$$\frac{W^+ - W^-}{W^-}. \quad (8)$$

A relação da diferença relativa a C é dada por:

$$\frac{W^+}{C} \text{ e } \frac{B^+}{C}, \quad (9)$$

$$\frac{W^-}{C} \text{ e } \frac{B^-}{C}.$$

5.2.2 Testes: Etapa 02

Para verificar o impacto no número de *transponders* quando usamos os algoritmos *STMB* e *STLB*, foram utilizadas as Equações (1) e (2). Já para identificar a quantidade de capacidade adicional requisitada para proteger a rede, o coeficiente de proteção de cada rede foi calculado utilizando a Equação (3) para o *STMB*, *STLB* e o *ST* para comparações.

5.2.3 Testes: Etapa 03

Nesta etapa, o processo de requisição e alocação de recursos apresentado no Capítulo 3 foi utilizado. Mas para a otimização da utilização de recursos, foi considerado o problema do Roteamento e Atribuição de Espectro (*RSA*). Aqui, o *RSA* foi introduzido em um cenário de rede dinâmica com esquema de proteção de caminho dedicado 1:1 para a realização de simulação de falhas. Como o *software ElasticO++* implementa *RMSA*, a modulação é fixa neste trabalho.

A base para o *RSA* foi uma combinação dos seguintes algoritmos: *STMB*, *STLB* e *ST* como algoritmos de roteamento, Deslocamento de Fase em Quadratura e Dupla Polarização (*EON DP – QPSK*) [Gerstel et al. 2012] como esquema de modulação fixo, *Spectrum Sharing* [Bonani et al. 2014] como técnica de gerenciamento e o *First Fit* como atribuição de espectro [Chatterjee, Sarma e Oki 2015]. Uma vez que o *DP – QPSK*, o *Spectrum Sharing* e o *First Fit* são frequentemente utilizados em publicações de algoritmos de atribuição, eles foram escolhidos para este trabalho.

O *DP – QPSK* é um esquema de modulação em que cada símbolo representa 4 bits [Gerstel et al. 2012]. O *Spectrum Sharing* é baseado na tecnologia *OFDM* e compartilha todo o espectro disponível de forma dinâmica para diferentes tipos de serviço, ocupando apenas a largura de banda suficiente para a transmissão [Bonani et al. 2014]. E o *First Fit* é uma política de alocação de espectro que mantém uma lista de índices disponíveis e *slots* usados. Para a atribuição de uma conexão, o *slot* indexado mais baixo da lista será definido. Quando a conexão for concluída, o *slot* é retornado à lista de *slots* disponíveis. A política de alocação do espectro *First Fit* é considerada uma das melhores políticas de alocação de espectro devido à menor probabilidade de bloqueio de chamadas e à complexidade da computação [Chatterjee, Sarma e Oki 2015].

A utilização de recursos é caracterizada pela taxa de bloqueio, R_b , definida como a relação de requisições bloqueadas com conexões interrompidas. Uma conexão será considerada “interrompida” quando esta estava em processo mas uma falha ocorreu antes dela finalizar. A equação que define R_b é dada por:

$$R_b = \frac{N_c - N_s + N_d}{N_c}. \quad (10)$$

onde N_c é o número total de requisições, N_s é o número total de requisições estabelecidas com sucesso, e N_d é o número total de conexões interrompidas. Quanto menor é a taxa de bloqueio, maior será a utilização do recurso [Wang, Zeng e Zhao 2003].

Como o objetivo era realizar uma simulação de falhas, um modelo foi necessário. O modelo de falha única do artigo [Wang, Zeng e Zhao 2003] foi utilizado pois possui características interessantes para este trabalho, empregando enlaces bidirecionais e parâmetros que lidam com falha única de ligação. Neste modelo, são considerados dois parâmetros não correlacionados: o intervalo entre falhas, T_i , que representa o tempo médio entre duas ocorrências consecutivas de falhas de enlace e a duração da falha, T_h , que representa o tempo médio que uma falha permanecerá.

A simulação de falhas prossegue deste modo: no tempo T_i a falha é lançada, e todas as conexões em serviço no enlace com falha serão quebradas. Este estado permanecerá até o tempo T_h . No tempo T_h é assumido que a falha já foi reparada. Assim permanecerá até o tempo T_{i+1} ocorrer e o processo se repete novamente. A Figura 19 apresenta esse processo de falhas.

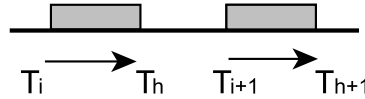


Figura 19 – Intervalo entre falhas e duração das falhas. Adaptado de [Wang, Zeng e Zhao 2003].

Em cada simulação executada, 10^5 requisições foram geradas. Cada nova requisição foi composta de uma origem, um destino e um requerimento de taxa de bit, seguindo uma distribuição uniforme. Quatro serviços foram providenciados pela rede com taxas de bits de 40 Gb/s, 100 Gb/s, 400 Gb/s e 1 Tb/s. Foram configurados 300 slots de frequência de 12,5 GHz como largura máxima de banda disponível por fibra (banda C). Ambos T_h e T_i tem valores fixos em cada simulação. Além disso, as simulações foram repetidas 30 vezes, cada uma com uma semente aleatória fixada.

Estas simulações foram divididas em três partes: (i) utilização de recursos versus a carga de tráfego, (ii) utilização de recursos versus o intervalo entre falhas, e (iii) utilização de recursos versus a duração da falha. Em (i), os valores para os parâmetros foram: $T_i = 1s$,

$T_h = 0,05$ s e λ (carga) = $\{50, 100, 150, \dots, 300\}$ Erlang. Em (ii): $T_i = \{0, 1, 0, 2, \dots, 10\}$ s, $T_h = 0,05$ s e $\lambda = 100$ Erlang. E em (iii): $T_i = 1$ s, $T_h = \{0, 1, 0, 2, \dots, 0, 9\}$ s e $\lambda = 100$ Erlang.

5.3 Conclusão

Apresentamos neste capítulo a metodologia de testes dos algoritmos para a obtenção de dados dos resultados. As métricas utilizadas e elaboradas para este trabalho foram exibidas. Já para a simulação de falhas foram apresentados os algoritmos definidos para a alocação de recursos com o *RSA* e o modelo de falhas, utilizados no esquema de proteção de caminho dedicado 1:1.

6 Resultados

Neste capítulo são apresentados os testes realizados com os algoritmos desenvolvidos. Na primeira parte, são identificados o comportamento e as diferenças com relação aos comprimentos dos caminhos dos algoritmos *STMB* e *STLB*. Na segunda parte, é verificado o impacto destes no número de *transponders* e no coeficiente de proteção da rede. Por último, uma simulação de falhas no esquema de proteção de caminho dedicado 1:1 é realizada. Após a apresentação dos testes, uma discussão referente a alguns resultados é abordada.

6.1 Cenário de Testes

Para testar os algoritmos, um conjunto de 40 redes de telecomunicações reais foi utilizado. Estas redes são *2-aresta-conexas* e são apresentadas na Tabela 2 [Routray et al. 2013, Pinto 2017], onde N representa o número de nodos, L representa o número de enlaces, $\langle \delta \rangle$ representa o grau nodal médio da rede ($\frac{2L}{N}$) e o tempo de processamento em segundos para executar cada rede utilizando os algoritmos *STMB* e o *STLB* é apresentado em *STMB* (s) e *STLB* (s).

Usando uma área de trabalho de PC equipada com uma CPU Intel Core i5-6600K @3.50GHz e 8GB de RAM, o total do tempo de processamento para calcular os caminhos utilizando o algoritmo do *STMB* em todas as 40 redes foi de aproximadamente 1 hora, a qual somente a maior rede (“USA”) gastou 57 minutos, e 3 minutos no total para as demais redes. O tempo de processamento obtido usando *STLB* foi maior, aproximadamente 7 horas somente para a rede “USA” e 3 minutos no total para as outras 39 redes.

A simulação de falhas foi baseada no artigo [Wang, Zeng e Zhao 2003], que realiza simulações extensas com base em seu modelo de falha de enlace. Nestes testes, foram comparados a utilização de recursos versus a carga de tráfego, o intervalo entre falhas e a duração da falha para o *STMB* e *STLB* sob demandas dinâmicas de tráfego nas topologias de rede “Arnes” e “Cernet” (Figura 20). A escolha destas redes se deu por apresentarem a diferença mais significativa entre o *STMB* e o *STLB*, resultados que são apresentados na sequência. O algoritmo *ST* também é comparado com os resultados do *STMB* e do *STLB* para uma melhor análise dos resultados. Neste trabalho, todos os enlaces são bidirecionais. Logo, uma falha em um enlace afeta ambas as direções.

Rede	N	L	$\langle\delta\rangle$	<i>STMB</i> (s)	<i>STLB</i> (s)
Vianet	9	12	2,67	0,10	0,10
Bren	10	11	2,20	0,11	0,10
Learn	10	11	2,20	0,12	0,10
Rnp	10	12	2,40	0,12	0,11
Abilenecore	10	13	2,60	0,11	0,11
Compuserv	11	14	2,55	0,15	0,14
Vbns	12	17	2,83	0,19	0,17
Cesnet	12	19	3,17	0,18	0,19
NSFNET	14	21	3,00	0,27	0,26
Italy	14	29	4,14	0,34	0,37
Mzima	15	19	2,53	0,34	0,32
Aconet	15	22	2,93	0,36	0,35
Hibernia	15	23	3,07	0,35	0,35
Garr-b	16	27	3,38	0,50	0,50
Arnes	17	20	2,35	0,51	0,47
Germany	17	26	3,06	0,61	0,58
Spain	17	28	3,29	0,58	0,58
Memorex	19	24	2,53	0,78	0,71
Canarie	19	26	2,74	0,83	0,78
Eon	19	37	3,89	0,93	0,96
Lambda	19	45	4,74	0,96	0,98
Sweden	20	24	2,40	0,92	0,82
Oxford	20	26	2,60	1,04	0,92
Arpanet	20	32	3,20	1,1	0,93
Pionier	21	25	2,38	1,09	1,04
Bulgaria	23	24	2,09	1,54	1,34
Cox	24	40	3,33	2,25	2,26
Sanet	25	28	2,24	2,16	1,84
Newnet	26	31	2,38	2,40	2,17
Portugal	26	36	2,77	2,71	2,61
Renater	27	35	2,59	3,22	2,99
Cernet	29	45	3,10	4,77	5,28
Geant2	30	48	3,20	5,68	5,68
IBN31	31	47	3,03	4,79	4,70
Loni	33	37	2,24	7,204	5,68
Metrona	33	41	2,48	8,97	7,78
Cost37	37	57	3,08	11,573	11,765
Omnicom	38	54	2,84	16,23	16,18
Internet2	56	61	2,18	50,98	44,33
USA	100	171	3,42	2725,22	25571,82

Tabela 2 – Conjunto de topologias de redes reais [Pinto 2017] e algumas de suas características: número de nodos e de enlaces, grau nodal médio, tempo de processamento em segundos no *STMB* e no *STLB*. As topologias estão ordenadas pelo número de nodos.

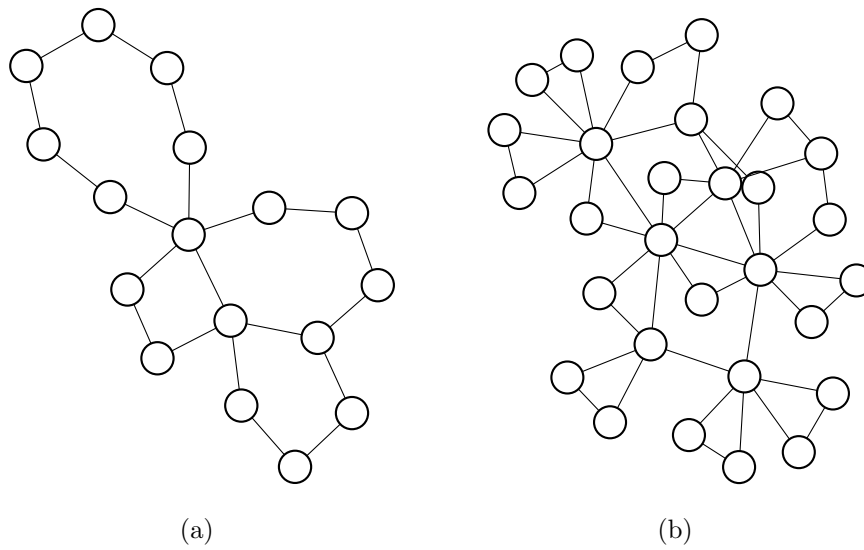


Figura 20 – (a) Rede Arnes, com 17 nodos e 20 enlaces; (b) Rede Cernet, com 29 nodos e 45 enlaces.

6.2 Comportamento e Diferenças do STMB e do STLB

Em ambos *STMB* e *STLB*, os comprimentos dos caminhos de trabalho e de *backup* foram obtidos para cada par de nodos origem e destino de cada topologia de rede. As diferenças entre comprimentos de caminhos de trabalho e caminhos de *backup* do *STMB* e do *STLB* para cada par de nodos origem e destino foram analisadas e subsequentemente, a soma dessas diferenças também. Então, os resultados obtidos para as 40 redes foram comparados.

Na primeira etapa de testes, é apresentada a diferença normalizada entre todas as 40 redes de telecomunicações reais. Após, a rede com maior diferença foi identificada e detalhada, e os resultados gerais de todas as redes considerando a soma das diferenças são apresentados na sequência.

Para a identificação das diferenças, as métricas apresentadas no Capítulo 5 são utilizadas para a obtenção de dados dos gráficos a seguir.

6.2.1 Diferença Normalizada do STMB e STLB nas 40 Redes de Telecomunicações Reais

Primeiramente, todas as redes foram analisadas com relação à diferença normalizada do comprimento dos caminhos de trabalho do *STMB* e do *STLB* e na Figura 21 os resultados são apresentados. É verificado que, os caminhos de trabalho do *STMB* serão iguais ou maiores até 29% que os caminhos de trabalho do *STLB*, sendo 7% na média. Os números fora do eixo acima de cada barra representam o total (de cada par origem e destino) do número de saltos utilizados a mais nos caminhos de trabalho definidos pelo

STMB. A rede que apresenta a maior diferença em porcentagem é a rede “Arnes”, com total de 118 saltos a mais nos caminhos de trabalho do *STMB*.

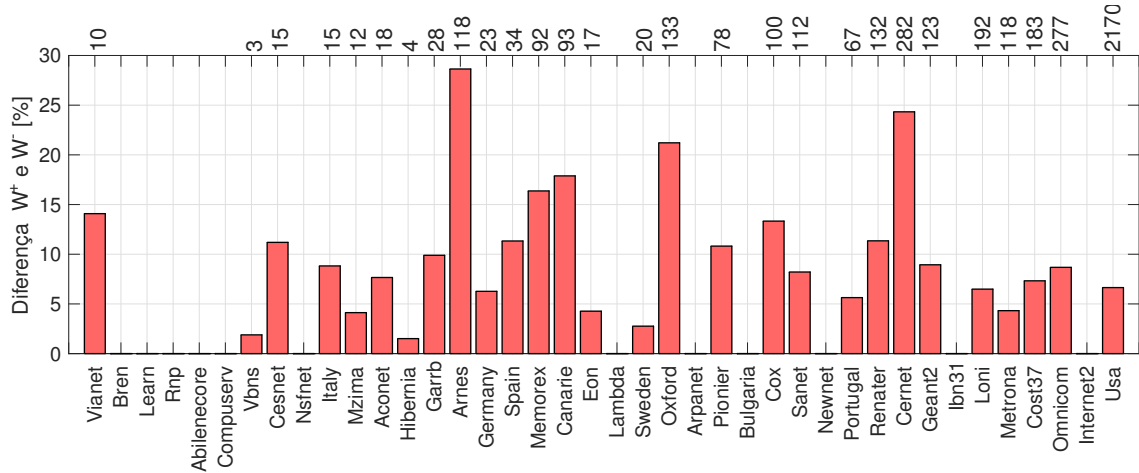


Figura 21 – Diferença normalizada dos caminhos de trabalho do *STMB* e do *STLB* para todas as redes. As redes estão organizadas pelo número de nodos em ordem crescente.

6.2.2 Diferenças do *STMB* e *STLB* Relativamente a C da Rede Arnes

A seguir, é apresentada uma investigação mais a fundo da rede “Arnes”, a qual apresentou a maior diferença entre as duas estratégias. Para isso, é identificado cada par de caminhos de cada par origem e destino.

Um dos pares de nodos que alcança a maior diferença entre o *STMB* e o *STLB* é o par (o, d) destacado na Figura 22, para o qual $C = 16$ arestas. No *STMB*, ambos caminhos de trabalho e de *backup* obtidos possuem 8 saltos, isto é, $W^+ = B^+ = 8$. No *STLB*, o caminho de trabalho obtido tem 3 saltos, $W^- = 3$, enquanto o caminho de *backup* tem 13 saltos, $B^- = 13$.

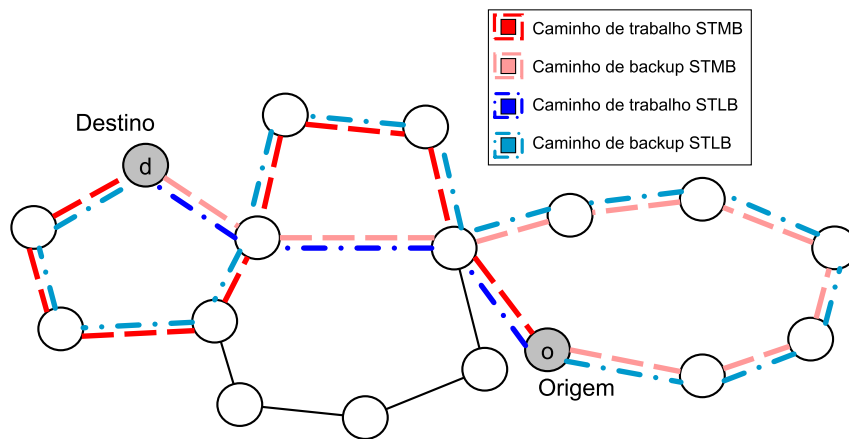


Figura 22 – Caminhos de trabalho e de *backup* definidos pelos algoritmos *STMB* e *STLB* na rede “Arnes” para os pares de nodos origem o e destino d .

Portanto, $\Theta_{o,d}^+ = 0$ e $\Theta_{o,d}^- = 10$ para este par. Além disso, como o caminho de trabalho do *STMB* tem 5 saltos a mais que o caminho de trabalho do *STLB* (consequentemente, o caminho de *backup* do *STLB* tem 5 saltos a mais que o caminho de *backup* do *STMB*), então $\Delta_{o,d} = 5$.

A rede “Arnes” tem 136 pares de nodos origem e destino. Desses, para 60% tem-se $\Theta_{o,d}^+ = \Theta_{o,d}^-$ e $\Delta_{o,d} = 0$, ou seja, os caminhos de trabalho e de *backup* definidos no *STMB* possuem o mesmo comprimento que os caminhos de trabalho e de *backup* definidos no *STLB*.

A Figura 23 apresenta os resultados de $\Theta_{o,d}^+$ e $\Theta_{o,d}^-$ para os outros 40% de pares origem e destino da rede “Arnes”. Nos resultados apresentados para cada par (o, d) , 100% corresponde à diferença do comprimento do caminho de trabalho com o comprimento do caminho de *backup* normalizados com C , que corresponde à soma do seus respectivos comprimentos de trabalho e de *backup*. Por exemplo, o $\Theta_{o,d}^- = 10$ mencionado anteriormente corresponde a 63% relativamente a $C = 16$. Grandes diferenças entre comprimentos de caminhos de trabalho e de *backup* levam a grandes valores de $\Theta_{o,d}^-$ e $\Theta_{o,d}^+$. Por outro lado, essas variáveis são minimizadas quando os caminhos de trabalho e de *backup* tem exatamente o mesmo comprimento. Além disso, $\Theta_{o,d}^- \geq \Theta_{o,d}^+$, como esperado.

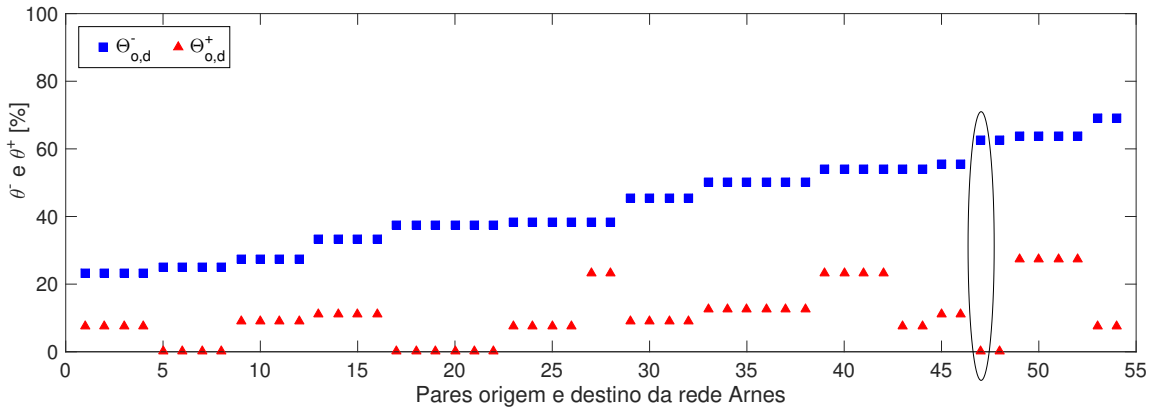


Figura 23 – Diferença normalizada entre os comprimentos dos caminhos de trabalho e de *backup* do *STMB* ($\Theta_{o,d}^+$) e *STLB* ($\Theta_{o,d}^-$) da rede “Arnes”. O eixo horizontal representa pares de origem e destino para os quais existe diferença nos comprimentos de caminhos entre as duas estratégias. O par apresentado na Figura 22 está circulado. Os pares estão organizados de acordo com $\Theta_{o,d}^-$ em ordem crescente.

A soma de todos os comprimentos de caminhos de trabalho do *STMB* e do *STLB* para a rede “Arnes” é $\sum W^+ = 530$ e $\sum W^- = 412$, respectivamente. Isso resulta em uma variação total de $\sum \Delta_{o,d} = 118$, ou seja, os dois algoritmos diferem 29% relativamente ao comprimento total do caminho de trabalho. Relativamente a $\sum C$ para todos os pares de nodos, a diferença alcança 9% (veja a Figura 24).

6.2.3 Diferenças do STMB e STLB Relativamente a C das 40 Redes de Telecomunicações Reais

Para comparar as diferenças entre caminhos de trabalho e de *backup* do *STMB* e *STLB* entre as 40 redes, primeiro foram calculados para cada rede: $\sum W^+$, $\sum W^-$, $\sum B^+$, $\sum B^-$. Então, $\sum \Theta_{o,d}^+$ e $\sum \Theta_{o,d}^-$ foram obtidos. O resultado é apresentado na Figura 24 onde, para cada rede, 100% corresponde ao $\sum C$ para todos os pares de nodos. Note que $\sum \Theta_{o,d}^- \geq \sum \Theta_{o,d}^+$, como esperado. No gráfico $\sum \Theta_{o,d}^-$ é representado por γ^- e $\sum \Theta_{o,d}^+$ por γ^+ .

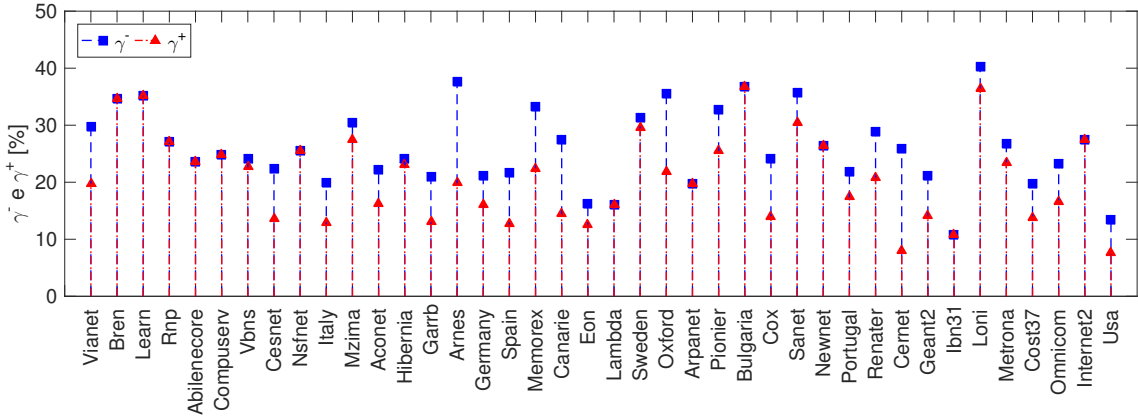


Figura 24 – Diferença normalizada entre a soma dos comprimentos dos caminhos de trabalho e a soma dos comprimentos dos caminhos de *backup* no *STMB* (γ^+) e *STLB* (γ^-). O eixo horizontal representa as 40 redes analisadas. As redes estão organizadas pelo seu número de nodos em ordem crescente.

Em 12 redes, os caminhos de trabalho e de *backup* definidos no *STMB* foram os mesmos definidos no *STLB* para cada par de nodos origem e destino. Essas redes tem $\sum \Theta_{o,d}^+ = \sum \Theta_{o,d}^-$. Portanto, não existe diferença ao utilizar *STMB* e *STLB* em 30% das redes analisadas.

Entre as redes para as quais $\sum \Theta_{o,d}^+ \neq \sum \Theta_{o,d}^-$, a rede “Loni” tem a maior diferença em comprimentos de caminhos de trabalho e de *backup* em ambos os algoritmos, com $\sum \Theta_{o,d}^- = 40\%$ e $\sum \Theta_{o,d}^+ = 36\%$, porcentagens relativas a $\sum C$. A menor diferença no *STMB* e no *STLB* é da rede “USA”, $\sum \Theta_{o,d}^- = 13\%$ e $\sum \Theta_{o,d}^+ = 8\%$ relativo a $\sum C$, significando que esta rede tem pares com a menor diferença em comprimento dos caminhos de trabalho e de *backup*.

A diferença entre os algoritmos foi obtida através da diferença dos caminhos de trabalho e de *backup* do *STMB* com a diferença dos caminhos de trabalho e de *backup* do *STLB*. Então, a porcentagem apresentada na Figura 24 é duas vezes a diferença entre as duas estratégias, isto é, 2Δ . As redes “Arnes” e “Cernet” obtiveram a maior diferença com $\sum \Theta_{o,d}^- - \sum \Theta_{o,d}^+ = 18\%$, isto é, $\Delta = 9\%$. Em média para todas as redes a diferença foi de 5%.

6.3 Impacto do STMB e STLB no Número de Transponders e no Coeficiente de Proteção

A Figura 25 apresenta o impacto no número de *transponders* através da diferença normalizada entre o número de *transponders* dos caminhos de trabalho calculados pelo *STMB* ($T(W^+)$) e pelo *STLB* ($T(W^-)$) para todas as redes. Entre as 40 redes, a rede “Arnes” é a que apresenta o maior impacto, utilizando até 29% mais *transponders* no *STMB* do que no *STLB*. Este resultado confirma a diferença esperada dos algoritmos apresentada na Seção 6.2.1. Em média para todas as redes testadas, o uso do *STMB* requer 7% mais *transponders*.

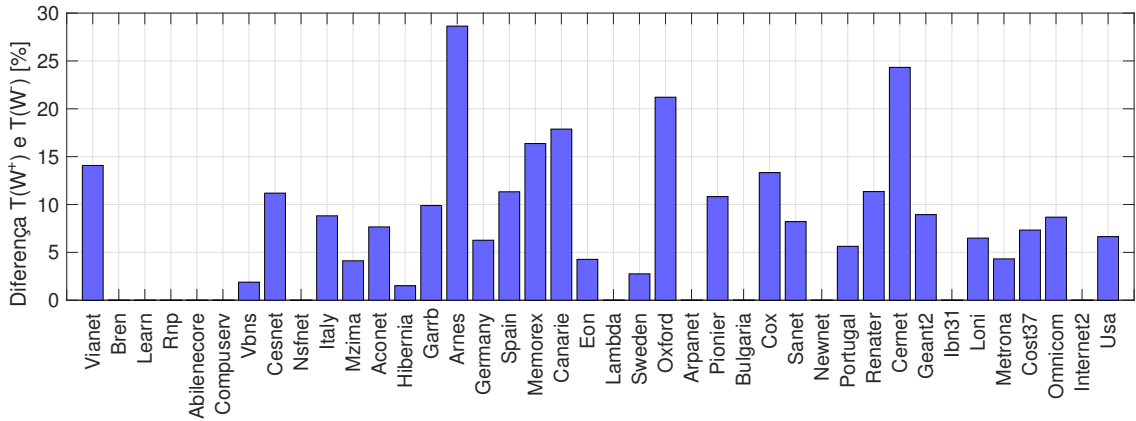


Figura 25 – Diferença normalizada entre o número de *transponders* usados nos caminhos de trabalho do *STMB* e *STLB* para todas as redes. As redes estão organizadas pelo número de nodos em ordem crescente.

A Figura 26 apresenta em porcentagem o coeficiente de proteção (k_p) de cada rede para o *STMB*, o *STLB* e o *ST*, ou seja, a quantidade de capacidade adicional requisitada para proteger a rede. Na Figura é verificado que a rede “Loni” precisa 235% de capacidade extra para garantir a sobrevivência contra falhas de enlace quando utiliza *STLB* e 214% utilizando *STMB*. Esta é a quantidade máxima de capacidade extra necessária dentre o conjunto de 40 redes excluindo aquelas que têm o mesmo comprimento de caminhos de trabalho e *backup* no *STMB* e *STLB*, isto é, considerando 28 redes. A capacidade extra mínima requerida é 131% no *STLB* e 116% no *STMB*, esses resultados são ambos da rede “USA”. Em média para todas as redes testadas, 173% é necessário no *STLB* e 155% no *STMB*. A maior diferença entre k_p do *STLB* e o k_p do *STMB* está na rede “Arnes”, para a qual o *STLB* requer 71% de capacidade extra para proteger a rede. O k_p no *STLB* apresenta resultados iguais ou maiores do que o k_p no *STMB*, significando que o *STLB* vai precisar mais capacidade extra na rede.

Considerando o *ST*, dentre as 28 redes com diferenças entre o comprimento de caminhos de trabalho e *backup* no *STMB* e no *STLB*, 20 redes obtiveram os mesmos

resultados do k_p do *STLB*. As outras, obtiveram resultados entre o k_p do *STLB* e o k_p do *STMB*, mas se aproximando mais do k_p do *STLB*.

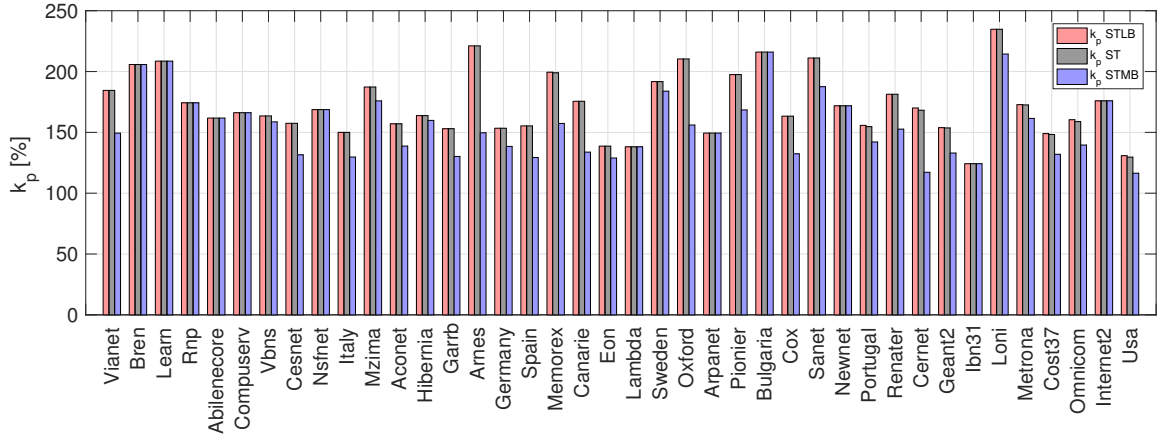


Figura 26 – Coeficiente de proteção (k_p) para cada rede usando os algoritmos *ST*, *STMB* e *STLB*.

Estes resultados implicam em um impacto significativo nos custos da rede, pois são obtidos uma redução de até 22% no custo de *transponders* ao usar *STLB* e uma redução de até 32% em capacidade extra de proteção ao usar *STMB*.

6.4 Simulação de Falhas: Utilização de Recursos versus Carga de Tráfego

Nas Figuras 27 (a) e 27 (b) a taxa de bloqueio (R_b) do *STMB*, *STLB* e *ST* é apresentada com uma função da taxa de chegada para as redes “Arnes” e “Cernet”, com intervalo entre falhas $T_i = 1$ s, duração da falha $T_h = 0,05$ s e λ (carga) = {50, 100, 150, ..., 300} Erlang.

Na rede “Arnes”, a utilização de recursos do *STMB* e *STLB* é baixa quando a carga do tráfego de rede é menor que 100 Erlang. Quando o tráfego aumenta, a taxa de bloqueio aumenta para todos os algoritmos. O *STMB* possui a maior taxa de bloqueio em todas as cargas, obtendo até 12% de diferença do *STLB*. Isto é, o *STLB* tem uma utilização de recursos melhor do que o *STMB*. O *ST* e o *STLB* apresentam mínima diferença, sendo o *ST* até 1,3% maior na carga 275 Erlang.

Para a rede “Cernet”, no tráfego leve a utilização de recursos é mínima até a carga 150 Erlang. Em seguida a taxa de bloqueio aumenta constantemente. Os algoritmos *STMB* e *STLB* possuem taxas de bloqueio quase idênticas, o *STMB* é somente 5,23% maior na carga 200 Erlang. Já o *ST* apresenta a maior taxa de bloqueio entre todos os algoritmos. Na carga 250 Erlang todos os algoritmos possuem quase a mesma taxa de bloqueio.

6.5 Simulação de Falhas: Utilização de Recursos versus Intervalo Entre Falhas

As Figuras 27 (c) e 27 (d) mostram a taxa de bloqueio versus o intervalo entre falhas do *STMB*, *STLB* e *ST* para ambas as redes, com $T_i = \{0, 1, 0, 2, \dots, 10\}$, $T_h = 0, 05$ e $\lambda = 100$. Quando T_i é menor que 1, a taxa de bloqueio é maior e muda rapidamente porque as falhas são frequentes. A medida que o T_i aumenta, a taxa de bloqueio estabiliza.

Na rede “Arnes” encontram-se grandes diferenças entre o *STMB* e o *STLB*, onde o *STMB* é até 92% maior em $T_i = 10, 0$. O *ST* se aproxima da taxa de bloqueio do *STLB* antes de $T_i = 1, 0$ e depois fica entre um meio termo entre o *STMB* e o *STLB*. Considerando a rede “Cernet”, o *STMB* obtém uma taxa de bloqueio maior que o *STLB* apenas após $T_i = 1, 0$, chegando a ser 49,3% maior que o *STLB* no $T_i = 5, 0$. O *ST* possui uma taxa de bloqueio bem acima do *STMB*, de até 101% em $T_i = 6, 0$, e do *STLB* até 163% em $T_i = 6, 0$. O *STLB* tem melhor desempenho relativo à utilização de recursos em ambas as redes.

6.6 Simulação de Falhas: Utilização de Recursos versus Duração da Falha

As Figuras 27 (e) e 27 (f) apresentam a taxa de bloqueio versus a duração da falha do *STMB*, *STLB* e *ST*, com $T_i = 1$, $T_h = \{0, 1, 0, 2, \dots, 0, 9\}$ e $\lambda = 100$.

As diferenças encontradas entre o *STMB* e o *STLB* variam até 59% no $T_h = 0, 0$ na rede “Arnes”, onde o *STMB* possui uma taxa de bloqueio maior e até 51% no $T_h = 0, 2$ na rede “Cernet”, onde o *STLB* possui uma taxa de bloqueio maior. Na rede “Arnes” o *ST* acompanha o *STLB*. Já na rede “Cernet” ele tem uma diferença considerável dos outros algoritmos, sendo até 127% maior que o *STMB* no $T_h = 0, 6$ e até 73% maior que o *STLB* no $T_h = 0, 6$, obtendo a maior taxa de bloqueio.

6.7 Tempo de Simulação da Rede USA no STMB e no STLB

A rede “USA” possui características topológicas que exigem mais verificações de comprimentos e de caminhos disjuntos por arestas, resultando num tempo maior de execução quando se utiliza o *STMB*. Já no *STLB*, o comportamento da rede é explicado através da Figura 24, onde é verificado que existem menos diferenças entre os comprimentos de caminhos de trabalho e de *backup* em grande parte dos pares origem e destino. Por conta disso, na execução do *STLB*, o *R* e o *S* (ver Capítulo 4) que iniciam com comprimentos com a maior diferença possível (a partir do menor caminho definido pelo *Dijkstra*) são

atualizados muitas vezes, como também muitas verificações de caminhos disjuntos por arestas ocorrem, justificando assim, o extenso tempo para simular a rede “USA” no *STLB*.

6.8 Taxa de Bloqueio do ST na Rede Cernet

Na Figura 27(f) foi identificado na rede “Cernet”, que a taxa de bloqueio é extremamente maior no *ST* em $T_h = 0,6$. Para entender o porque a taxa de bloqueio demonstrou este comportamento, foram realizadas verificações de caminhos, de bloqueios de contiguidade, continuidade e de falta de recursos e também outras simulações. Os resultados referentes a estas verificações são apresentados a seguir.

A rede “Cernet” tem 29 nodos, logo possui 406 pares de nodos. Considerando dois caminhos para cada par de nodos, um de trabalho e um de *backup*, se tem um total de 812 caminhos. Sabendo disso, primeiramente foram analisados os caminhos que o *ST*, *STMB* e *STLB* retornam para cada par de nodos origem e destino da rede “Cernet”, na situação onde se encontra a maior diferença de taxa de bloqueio do *ST* com relação à do *STMB* e do *STLB* ($\lambda = 100$, $T_i = 1,0$, $T_h = 0,6$, e 10^5 requisições) e, considerando o T_h , a situação com a menor diferença também foi analisada ($\lambda = 100$, $T_i = 1,0$, $T_h = 0,0$, e 10^5 requisições).

Para a situação com a maior diferença de taxa de bloqueio do *ST* com relação à do *STMB* e do *STLB*, foram identificados os caminhos do *ST* que são iguais ou diferentes dos caminhos do *STMB* e do *STLB*. Então:

- Para 5,05% dos pares de nodos, os caminhos usados no *ST* foram os mesmos usados no *STMB*;
- Para 53,60% dos pares de nodos, os caminhos usados no *ST* foram os mesmos usados no *STLB*;
- Para 24,50% dos pares de nodos, os caminhos usados no *ST* foram os mesmos usados no *STMB* e no *STLB*;
- Para 16,85% dos pares de nodos, os caminhos usados no *ST* não foram os mesmos usados no *STMB* e no *STLB*. Considerando este caso de caminhos do *ST* diferentes dos caminhos do *STMB* e do *STLB*:
 - Para 28,47% dos pares de nodos, os comprimentos dos caminhos usados no *ST* foram os mesmos usados no *STLB*;
 - Para 71,53% dos pares de nodos, os comprimentos dos caminhos usados no *ST* foram os mesmos usados no *STMB* e no *STLB*;

Considerando os 24,50% dos pares de nodos com caminhos do *ST* que foram os mesmos usados no *STMB* e no *STLB*:

- Somando com os 5,05% dos pares de nodos com caminhos do *ST* que foram os mesmos usados no *STMB*: se tem aproximadamente 30% de pares de nodos com caminhos do *ST* que são os mesmos do *STMB*;
- Somando com os 53,60% dos pares de nodos com caminhos do *ST* que foram os mesmos usados no *STLB*: se tem aproximadamente 78% de pares de nodos com caminhos do *ST* que são os mesmos do *STLB*.

Para a situação com a menor diferença de taxa de bloqueio do *ST* com relação à do *STMB* e do *STLB*, foram identificados os caminhos do *ST* que são iguais ou diferentes dos caminhos do *STMB* e do *STLB*. Então:

- Para 4,06% dos pares de nodos, os caminhos usados no *ST* foram os mesmos usados no *STMB*;
- Para 59,85% dos pares de nodos, os caminhos usados no *ST* foram os mesmos usados no *STLB*;
- Para 24,02% dos pares de nodos, os caminhos usados no *ST* foram os mesmos usados no *STMB* e no *STLB*;
- Para 12,07% dos pares de nodos, os caminhos usados no *ST* não foram os mesmos usados no *STMB* e no *STLB*. Considerando este caso de caminhos do *ST* diferentes dos caminhos do *STMB* e do *STLB*:
 - Para 4,08% dos pares de nodos, os comprimentos dos caminhos usados no *ST* foram os mesmos usados no *STMB*;
 - Para 56,12% dos pares de nodos, os comprimentos dos caminhos usados no *ST* foram os mesmos usados no *STLB*;
 - Para 39,80% dos pares de nodos, os comprimentos dos caminhos usados no *ST* foram os mesmos usados no *STMB* e no *STLB*;

Considerando os 24,02% dos pares de nodos com caminhos do *ST* que foram os mesmos usados no *STMB* e no *STLB*:

- Somando com os 4,06% dos pares de nodos com caminhos do *ST* que foram os mesmos usados no *STMB*: se tem aproximadamente 28% de pares de nodos com caminhos do *ST* que são os mesmos do *STMB*;

- Somando com os 59,85% dos pares de nodos com caminhos do ST que foram os mesmos usados no $STLB$: se tem aproximadamente 84% de pares de nodos com caminhos do ST que são os mesmos do $STLB$.

As Figuras 28(a), 28(b), 28(c) e 28(d) resumem os dados apresentados referentes aos caminhos e comprimentos do ST iguais ao $STMB$ e/ou ao $STLB$ considerando a maior e menor diferença entre as taxas de bloqueios no T_h dos algoritmos ST , $STMB$ e $STLB$.

Como estes resultados não foram suficientes para entender a taxa de bloqueio do ST , novas verificações foram necessárias.

Em seguida, foram analisadas as taxas de bloqueios referentes as restrições (continuidade e contiguidade) e por falta de recursos. Considerando a situação onde se encontra a maior diferença de taxa de bloqueio do ST com relação à do $STMB$ e do $STLB$ ($\lambda = 100$, $T_i = 1, 0$, $T_h = 0, 6$, e 10^5 requisições), os valores referentes as taxas de bloqueios e as taxas de bloqueios de cada tipo de bloqueio estão resumidos na Tabela 3. Na Figura 29 estão apresentadas as taxas de bloqueio de contiguidade e de continuidade.

Algoritmo	Taxa de Bloqueio	Contiguidade	Continuidade	Falta de Recursos
ST	0,0000976667	0,649531049	0,346302284	0
STMB	0,0000590000	0,399549062	0,600450938	0
STLB	0,0000676667	0,643276103	0,356723897	0

Tabela 3 – Taxas de bloqueio no ST , no $STMB$ e no $STLB$ e taxas de bloqueio de tipos de bloqueio: de contiguidade, de continuidade e de falta de recursos.

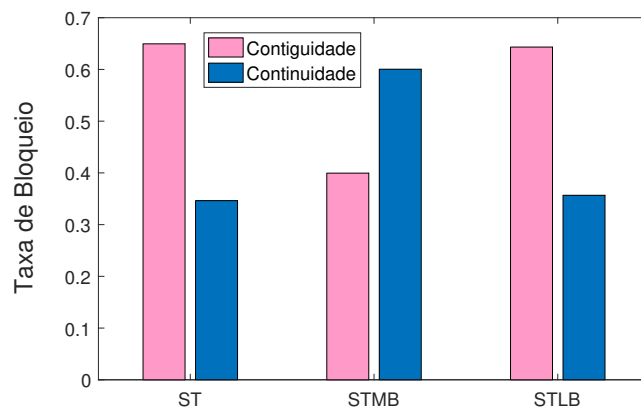


Figura 29 – Taxas de bloqueios de contiguidade e continuidade do ST , $STMB$ e $STLB$.

Mais uma vez, não foram identificadas razões para a taxa de bloqueio do ST ser extremamente maior.

Por fim, foram consideradas simulações com carga diferente, com $\lambda = 200$, $T_i = 1, 0$, $T_h = 0, 6$, e 10^5 requisições. As taxas de bloqueio estão resumidas na Tabela 4 e apresentadas na Figura 30.

Algoritmo	Taxa de Bloqueio
ST	0,033634000
STMB	0,031006000
STLB	0,030605333

Tabela 4 – Taxas de bloqueio no *ST*, no *STMB* e no *STLB* com $\lambda = 200$, $T_i = 1, 0$, $T_h = 0, 6$, e 10^5 requisições.

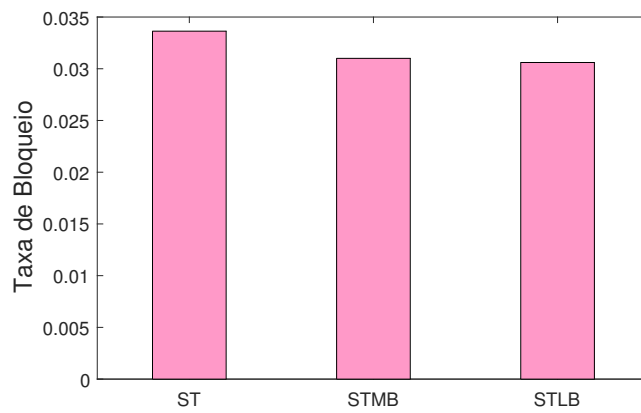


Figura 30 – Taxas de bloqueios do *ST*, *STMB* e *STLB* referentes a $\lambda = 200$, $T_i = 1, 0$, $T_h = 0, 6$, e 10^5 requisições.

E considerando a carga utilizada nas simulações anteriores, porém, com 10^6 requisições para $\lambda = 100$, $T_i = 1, 0$ e $T_h = 0, 6$, as taxas de bloqueio estão resumidas na Tabela 5 e apresentadas na Figura 31.

Algoritmo	Taxa de Bloqueio
ST	0,0000963000
STMB	0,0000589000
STLB	0,0000710333

Tabela 5 – Taxas de bloqueio no *ST*, no *STMB* e no *STLB* com $\lambda = 100$, $T_i = 1, 0$ e $T_h = 0, 6$, e 10^6 requisições.

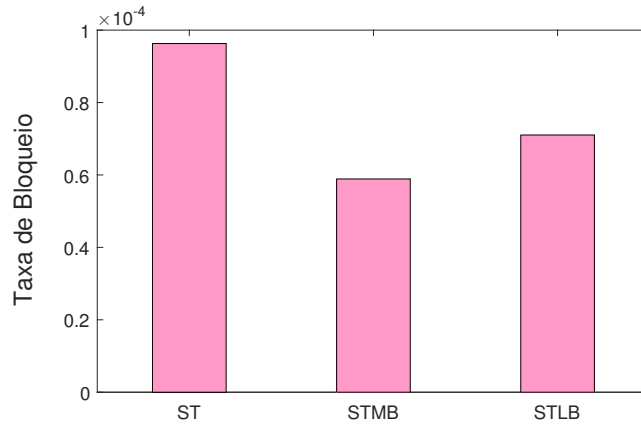


Figura 31 – Taxas de bloqueios do *ST*, *STMB* e *STLB* referentes a $\lambda = 100$, $T_i = 1, 0$, $T_h = 0, 6$, e 10^6 requisições.

Estas taxas de bloqueios apresentaram as mesmas tendências de resultados já encontradas.

Considerando os resultados destas verificações, conclui-se que os aproximados 17% de caminhos do *ST* que são diferentes dos caminhos do *STMB* e do *STLB*, possam ser o motivo desta grande diferença na taxa de bloqueio do algoritmo *ST* em comparação a taxa de bloqueio do *STMB* e do *STLB*. Porém, novas pesquisas devem ser realizadas em trabalhos futuros para identificar melhor este resultado e obter novas respostas.

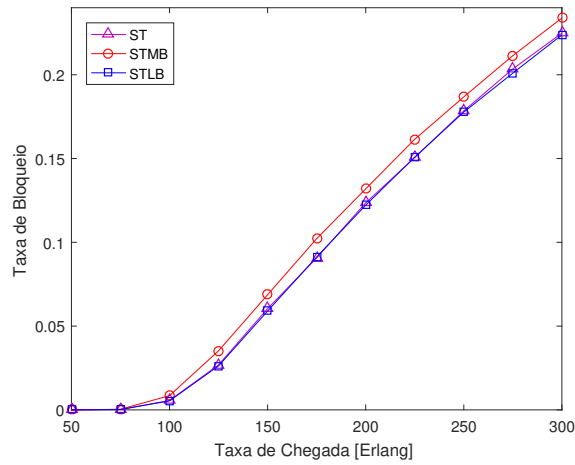
6.9 Conclusão

Este capítulo apresentou as principais diferenças com relação a comprimentos de caminhos do Suurballe e Tarjan Mais Balanceado (*STMB*) e do Suurballe e Tarjan Menos Balanceado (*STLB*), como também o seu impacto no número de *transponders* e no coeficiente de proteção de 40 redes de telecomunicações reais. Também, exibiu resultados de uma simulação de falhas no cenário de rede de proteção de caminho dedicado 1:1. Além disso, apresentou discussões referentes a resultados da rede “USA” e da rede “Cernet”.

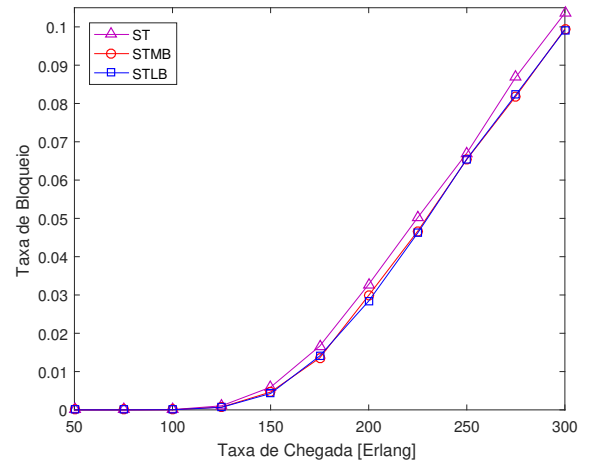
Dentre as redes testadas, foram identificadas as redes “Arnes” e “Cernet” com a maior diferença 9%, encontrada entre os algoritmos (porcentagem relativa à soma dos comprimentos dos caminhos de trabalho e de *backup* para todos os pares de nodos). Já com relação ao número de *transponders*, a rede “Arnes” é a que mais tem impacto dentre as outras redes, com uma necessidade de 29% a mais de *transponders* utilizando o *STMB*, em comparação ao *STLB*. No coeficiente de proteção, a rede “Loni” apresentou 235% de capacidade extra necessária para manter a rede protegida quando se utiliza o *STLB*, em comparação ao *STMB*.

Na simulação de falhas, o *STLB* obteve a melhor utilização na rede “Arnes”, pois o *STMB* obteve uma taxa de bloqueio até 12% maior. Já na rede “Cernet”, o *STMB* e o

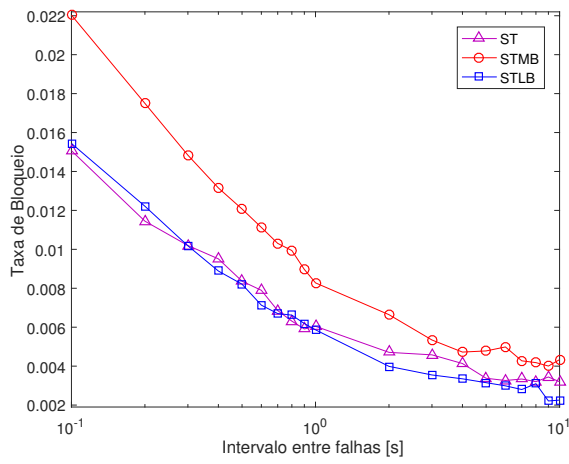
STLB tiveram taxas de bloqueio próximas. Porém diferenças na utilização de intervalo entre falhas e na duração da falha foram identificadas. No primeiro caso, o *STLB* obteve um pior desempenho quanto à utilização de recursos quando o T_i é menor que 1,0. Já com T_i maior que 1,0, os resultados mudam e o *STMB* chega a ser até 49% maior que o *STLB*. Para o caso da duração da falha, foram apresentados valores em que o T_h varia de 0,1 à 0,9, portanto, é esperado que o *STLB* tenha um comportamento pior na utilização de recursos, porque a duração da falha é maior. Logo, a taxa de bloqueio do *STLB* é até 51% maior que no *STMB*. Estes dados sugerem que quando houver muitas falhas e a duração for maior, o *STMB* terá um melhor desempenho. Considerando o *ST*, nas simulações na rede “Arnes”, ele obteve taxas de bloqueio aproximadas das taxas de bloqueio do *STLB*. Já na rede “Cernet”, as taxas de bloqueio do *ST* foram superiores ao do *STMB* e do *STLB*, sendo até 163% maior que o *STLB* quando o intervalo entre falhas está em $T_i = 6,0$ e 127% maior que o *STMB* quando a duração da falha é $T_h = 0,6$.



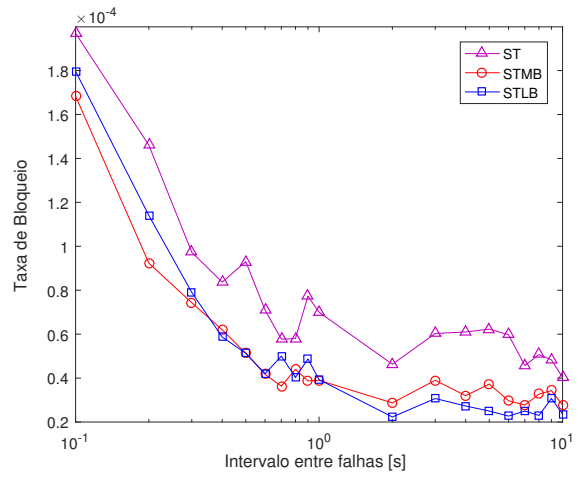
(a) Arnes



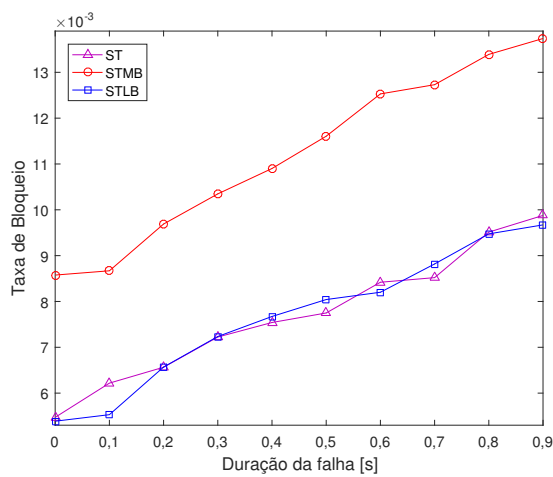
(b) Cernet



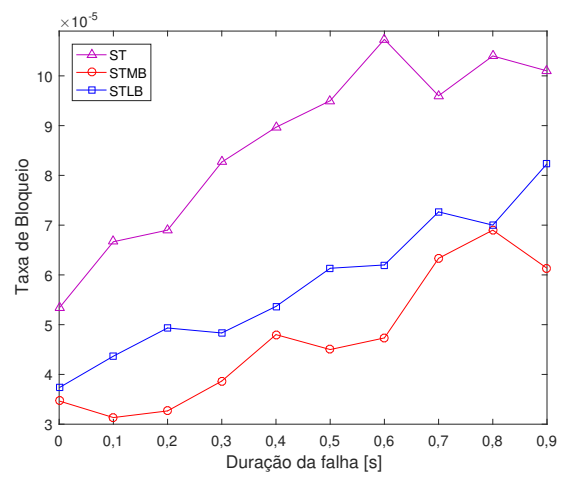
(c) Arnes



(d) Cernet



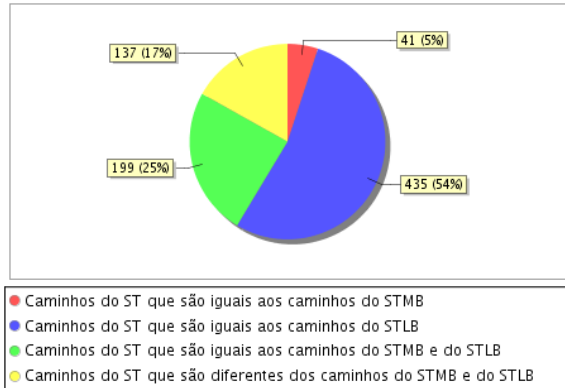
(e) Arnes



(f) Cernet

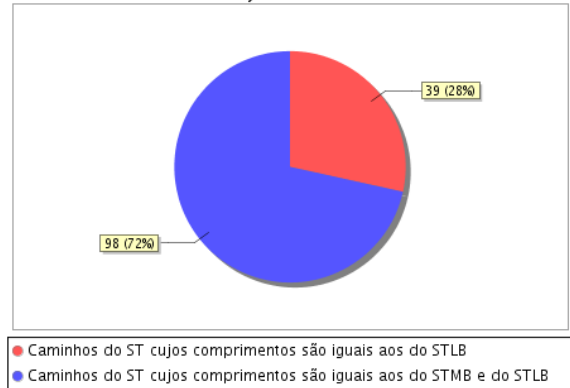
Figura 27 – Taxa de bloqueio vs. taxa de chegada das redes Arnes (a) e Cernet (b), para $T_i = 1$ e $T_h = 0,05s$. Taxa de bloqueio vs. intervalo entre falhas das redes Arnes (c) e Cernet (d), para $\lambda = 100$ e $T_h = 0,05s$. Taxa de bloqueio vs. duração da falha das redes Arnes (e) e Cernet (f), para $\lambda = 100$ e $T_i = 1s$.

Coincidência de caminhos entre o ST, o STMB e o STL



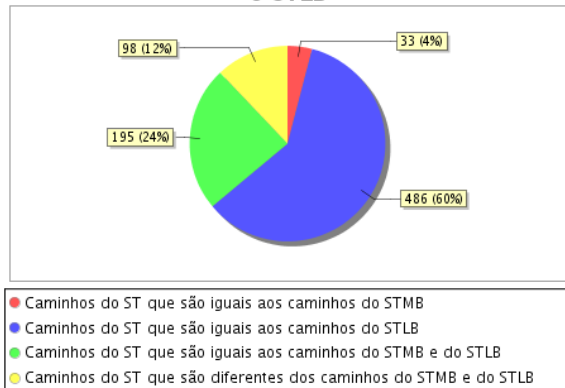
(a)

Coincidência de comprimentos de caminhos entre o ST, o STMB e o STL



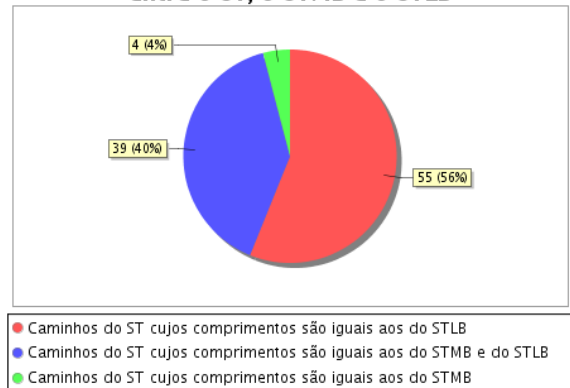
(b)

Coincidência de caminhos entre o ST, o STMB e o STL



(c)

Coincidência de comprimentos de caminhos entre o ST, o STMB e o STL



(d)

Figura 28 – Considerando a maior diferença de taxa de bloqueio do *ST* com relação à do *STMB* e do *STLB*: (a) Coincidência de caminhos entre o *ST*, o *STMB* e o *STLB*; (b) Coincidência de comprimentos de caminhos entre o *ST*, o *STMB* e o *STLB*; Considerando a menor diferença de taxa de bloqueio do *ST* com relação à do *STMB* e do *STLB*: (c) Coincidência de caminhos entre o *ST*, o *STMB* e o *STLB*; (d) Coincidência de comprimentos de caminhos entre o *ST*, o *STMB* e o *STLB*.

7 Conclusão e Trabalhos Futuros

Duas versões do algoritmo *Suurballe e Tarjan (ST)* foram propostas neste trabalho com o objetivo de explorar a diversidade de comprimentos de caminhos de trabalho e de *backup* encontradas em um mesmo par de nodos origem e destino em uma topologia. As versões foram chamadas de Suurballe e Tarjan Mais Balanceado (*STMB*) e Suurballe e Tarjan Menos Balanceado (*STLB*). O *STMB* encontra o menor par de caminhos de trabalho e de *backup* com comprimentos com a menor diferença possível, enquanto que o *STLB* encontra a combinação com comprimentos com a maior diferença possível. Os algoritmos *STMB* e *STLB* permitem a escolha de comprimentos de caminhos de trabalho e de *backup*, tornando possível explorar diferentes estratégias de proteção para aplicações em diferentes cenários.

Os algoritmos propostos foram testados em um conjunto de 40 redes ópticas de telecomunicações reais. Para cada par de nodos origem e destino de cada rede, a diferença entre os comprimentos de caminhos de trabalho e de *backup* do *STMB* e do *STLB* foram explorados. Subsequentemente, a soma dessas diferenças para todos os pares de nodos foi analisada.

Os resultados mostraram que pode existir grandes diferenças entre os comprimentos de caminhos de trabalho e de *backup* nos dois algoritmos. No caso do *STLB*, a diferença entre esses comprimentos pode chegar a até 63% para um par de nodos origem e destino da rede “Arnes”. Diferenças entre o *STMB* e o *STLB* também foram observadas em 70% das redes analisadas, onde foi encontrado até 9% de diferença entre os algoritmos nas redes “Arnes” e “Cernet”.

Esta diferença observada impacta o coeficiente de proteção e o número de *transponders* requisitados pelos esquemas de proteção de caminho dedicado. A rede “Arnes” precisa de até 29% mais *transponders* quando se usa o *STMB* do que quando se usa o *STLB* e requer 71% mais capacidade de proteção com o *STLB* do que com o *STMB*. A rede que mais necessita de capacidade extra em ambos os algoritmos é a rede “Loni”, a qual precisa 235% quando utiliza o *STLB* e 214% com o *STMB*.

As versões propostas podem ser usadas em diferentes tipos de serviço. Se a necessidade for caminhos com comprimentos iguais ou com a menor diferença possível, a melhor estratégia é usar o *STMB*. Assim, poderá ser obtida uma redução de até 32% em capacidade de proteção extra. Entretanto, o *STMB* pode requerer utilização de mais *transponders*. Se a necessidade for que o caminho de trabalho seja o menor possível, a estratégia mais apropriada é o algoritmo *STLB*. Assim, poderá ser obtida uma redução de até 22% em custos de *transponders*. No entanto, a capacidade extra para proteger a

rede em caso de falha pode ser maior.

Além das diferenças entre os algoritmos e o impacto no número de *transponders* e na capacidade de proteção da rede, também foi investigado o comportamento dos algoritmos *STMB* e *STLB* em redes ópticas elásticas através de simulações nas topologias de redes “Arnes” e “Cernet”, considerando proteção de caminho dedicado 1:1 e falha única de enlace. Foram avaliados o impacto da carga de tráfego, o intervalo de tempo entre falhas de enlace e a duração da falha na utilização de recursos, considerando um cenário de tráfego dinâmico. Juntamente com os algoritmos *STMB* e *STLB* foram apresentados os resultados para o algoritmo *ST*.

Todas as simulações de cenário com falhas na rede “Arnes” apresentaram melhor utilização de recursos ao implementar o *STLB*. O *STMB* obteve uma taxa de bloqueio até 12% maior. Esse comportamento pode ser explicado pelo fato de que os comprimentos de caminhos de trabalho do *STLB* são menores, utilizando menos recursos antes de uma falha. Para o caso do *ST*, este algoritmo obteve resultados aproximados ao *STLB*.

Nas simulações para a rede “Cernet”, o *STMB* e o *STLB* demonstraram serem melhores que o *ST*, pois este apresentou uma taxa de bloqueio maior. Quanto ao *STMB* e ao *STLB* na taxa de bloqueio, não houve diferenças significativas. Entretanto, no intervalo entre falhas (T_i), o *STLB* obteve um pior desempenho quanto à utilização de recursos quando o T_i é menor que 1,0. Já com T_i maior que 1,0, os resultados mudam e a taxa de bloqueio no *STMB* chega a ser até 49% maior que no *STLB*. Estes dados sugerem que quando houver muitas falhas, o *STMB* terá um melhor desempenho, pois os comprimentos de ambos seus caminhos possuem a menor diferença possível, utilizando uma mesma quantidade de recursos quando houver falhas.

A diversidade de comprimentos de caminhos de trabalho e de *backup* encontradas em um mesmo par de nodos origem e destino em uma topologia pode impactar os custos da rede. Os resultados apresentados das duas redes de telecomunicações reais “Arnes” e “Cernet”, demonstram que a melhor opção para uso do algoritmo de roteamento, sendo *STMB* ou *STLB*, depende muito da topologia física da rede. Portanto no planejamento, é válido executar simulações com ambos os algoritmos para identificar o mais adequado para o cenário de rede.

Como trabalhos futuros tem-se: comparar o *STMB* e o *STLB* com outros algoritmos de roteamento; propor novos algoritmos baseados no *STMB* e *STLB*; analisar o *STMB* e *STLB* considerando esquemas de proteção de caminho compartilhado; identificar características no grafo (como medidas de centralidade, coeficiente de agrupamento, grau nodal, etc...) que relacionem os resultados apresentados e possam prever as diferenças encontradas; e testar outros algoritmos, como A^* ao invés do *Dijkstra* e *2-fluxo* ao invés do *Suurballe* e *Tarjan*.

Referências

- AMAYA, N.; ZERVAS, G.; SIMEONIDOU, D. Introducing node architecture flexibility for elastic optical networks. *IEEE/OSA Journal of Optical Communications and Networking*, IEEE, v. 5, n. 6, p. 593–608, 2013. Citado na página [12](#).
- ARORA, S.; LEE, H.; THURIMELLA, R. *Algorithms for finding disjoint paths in mobile networks*. [S.l.]: Citeseer, 2003. Citado na página [4](#).
- ASSI, C. et al. A hybrid distributed fault-management protocol for combating single-fiber failures in mesh-based DWDM optical networks. In: IEEE. *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE*. [S.l.], 2002. v. 3, p. 2676–2680. Citado na página [6](#).
- BERGER, L. et al. GMPLS segment recovery. 2007. Citado na página [17](#).
- BOAVENTURA-NETTO, P. Grafos: Teoria, modelos, algoritmos, 4th. *Edition*, Edgard Blücher, São Paulo, 2006. Citado na página [2](#).
- BONANI, L. H. et al. Routing issues on spectrum sharing and partitioning for flexible optical networks. In: IEEE. *Transparent Optical Networks (ICTON), 2014 16th International Conference on*. [S.l.], 2014. p. 1–4. Citado na página [41](#).
- BOUILLET, E. et al. *Path routing in mesh optical networks*. [S.l.]: John Wiley & Sons, 2007. Citado na página [10](#).
- CHAN, W.; CHIN, F. Y. Efficient algorithms for finding disjoint paths in grids. In: SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*. [S.l.], 1997. Citado na página [4](#).
- CHATTERJEE, B. C.; SARMA, N.; OKI, E. Routing and spectrum allocation in elastic optical networks: A tutorial. *IEEE Communications Surveys & Tutorials*, IEEE, v. 17, n. 3, p. 1776–1800, 2015. Citado 9 vezes nas páginas [1](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#), [15](#) e [41](#).
- CHRISTODOULOPOULOS, K.; TOMKOS, I.; VARVARIGOS, E. Elastic bandwidth allocation in flexible OFDM-based optical networks. *Journal of Lightwave Technology*, IEEE, v. 29, n. 9, p. 1354–1366, 2011. Citado 6 vezes nas páginas [viii](#), [1](#), [9](#), [10](#), [13](#) e [14](#).
- DAHSHAN, M. H. An algorithm for finding optimal node-disjoint paths in optical switched networks. In: IEEE. *Computer Technology and Development (ICCTD), 2010 2nd International Conference on*. [S.l.], 2010. p. 47–51. Citado na página [4](#).
- DAHSHAN, M. H. Maximum-bandwidth node-disjoint paths. *arXiv preprint arXiv:1305.0907*, 2013. Citado na página [4](#).
- DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische mathematik*, Springer, v. 1, n. 1, p. 269–271, 1959. Citado 2 vezes nas páginas [25](#) e [35](#).
- DINNEEN, M. J.; KIM, Y.-B.; NICOLESCU, R. Edge-and node-disjoint paths in p systems. *arXiv preprint arXiv:1011.0495*, 2010. Citado na página [4](#).

- FAWAZ, W. et al. A novel fault management approach for DWDM optical networks. *International Journal of Network Management*, Wiley Online Library, v. 16, n. 5, p. 337–349, 2006. Citado na página 6.
- GERSTEL, O. et al. Elastic optical networking: A new dawn for the optical layer? *IEEE Communications Magazine*, IEEE, v. 50, n. 2, 2012. Citado 4 vezes nas páginas 9, 13, 23 e 41.
- GIROLIMETTO, M.; PAVAN, C. *The Average Number of Hops for Working and Backup Paths in Telecommunication Networks*. [S.l.]: SSN, 2016. Citado 2 vezes nas páginas 3 e 4.
- GIROLIMETTO, M. et al. Exploiting the diversity of shortest pairs of edge-disjoint paths. *Simpósio Brasileiro de Redes de Computadores (SBRC)*, v. 36, 2018. Disponível em: <<http://portaldeconteudo.sbc.org.br/index.php/sbrc/article/view/2473>>. Citado na página 7.
- GOSCIEN, R. et al. Protection in elastic optical networks. *IEEE Network*, IEEE, v. 29, n. 6, p. 88–96, 2015. Citado 2 vezes nas páginas 3 e 17.
- ILYAS, M.; MOUFTAH, H. T. *The handbook of optical communication networks*. [S.l.]: CRC Press, 2003. Citado 2 vezes nas páginas 16 e 17.
- JINNO, M. et al. Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies. *IEEE communications magazine*, IEEE, v. 47, n. 11, 2009. Citado 2 vezes nas páginas 10 e 12.
- KOROTKY, S. K. Network global expectation model: A statistical formalism for quickly quantifying network needs and costs. *Journal of Lightwave Technology*, IEEE, v. 22, n. 3, p. 703, 2004. Citado 2 vezes nas páginas 6 e 18.
- KUIPERS, F. A. An overview of algorithms for network survivability. *ISRN Communications and Networking*, Hindawi Publishing Corporation, v. 2012, 2012. Citado na página 4.
- LABOURDETTE, J.-F. et al. Fast approximate dimensioning and performance analysis of mesh optical networks. *IEEE/ACM Transactions on Networking (TON)*, IEEE Press, v. 13, n. 4, p. 906–917, 2005. Citado 2 vezes nas páginas 6 e 18.
- LANG, J.; REKHTER, Y.; PAPADIMITRIOU, D. *RSVP-TE extensions in support of end-to-end generalized multi-protocol label switching (GMPLS) recovery*. [S.l.], 2007. Citado na página 17.
- LI, C.-L.; MCCORMICK, S. T.; SIMCHI-LEVI, D. Finding disjoint paths with different path-costs: Complexity and algorithms. *Networks*, Wiley Online Library, v. 22, n. 7, p. 653–667, 1992. Citado na página 4.
- MANCHESTER, J.; BONENFANT, P.; NEWTON, C. The evolution of transport network survivability. *IEEE Communications Magazine*, IEEE, v. 37, n. 8, p. 44–51, 1999. Citado na página 5.
- MANDELBAUM, A.; ZELTYN, S. The palm/erlang-a queue, with applications to call centers. In: *Technical Report*. [S.l.]: Service Engineering Lecture Notes, 2004. Citado na página 23.

- MARTINS, E. Q.; PASCOAL, M. M. A new implementation of Yen's ranking loopless paths algorithm. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, Springer, v. 1, n. 2, p. 121–133, 2003. Citado 3 vezes nas páginas [viii](#), [28](#) e [29](#).
- NYKÄNEN, M.; UKKONEN, E. The exact path length problem. *Journal of Algorithms*, Elsevier, v. 42, n. 1, p. 41–53, 2002. Citado na página [38](#).
- OLIVEIRA, J. M. S. d. S. *Protecção máxima de redes de telecomunicações*. Dissertação (Mestrado) — Universidade de Aveiro, 2010. Citado na página [25](#).
- PARK, J.-H.; CHOI, J.; LIM, H.-S. Algorithms for finding disjoint path covers in unit interval graphs. *Discrete Applied Mathematics*, Elsevier, v. 205, p. 132–149, 2016. Citado na página [4](#).
- PATEL, A. N. et al. Routing, wavelength assignment, and spectrum allocation algorithms in transparent flexible optical wdm networks. *Optical Switching and Networking*, Elsevier, v. 9, n. 3, p. 191–204, 2012. Citado na página [9](#).
- PAVAN, C. et al. How reliable are the real-world optical transport networks? *Journal of Optical Communications and Networking*, Optical Society of America, v. 7, n. 6, p. 578–585, 2015. Citado 2 vezes nas páginas [3](#) e [18](#).
- PINTO, A. N. *Reference Networks*. 2017. www.av.it.pt/anp/on/refnet2.html. Citado 4 vezes nas páginas [xi](#), [6](#), [44](#) e [45](#).
- RAMAMURTHY, B. et al. Transparent vs. opaque vs. translucent wavelength-routed optical networks. In: OPTICAL SOCIETY OF AMERICA. *Optical Fiber Communication Conference*. [S.l.], 1999. p. TuF2. Citado na página [10](#).
- RAMAMURTHY, S.; SAHASRABUDDHE, L.; MUKHERJEE, B. Survivable WDM mesh networks. *Journal of Lightwave Technology*, IEEE, v. 21, n. 4, p. 870, 2003. Citado 2 vezes nas páginas [3](#) e [16](#).
- RAMASWAMI, R.; SIVARAJAN, K.; SASAKI, G. *Optical networks: a practical perspective*. [S.l.]: Morgan Kaufmann, 2009. Citado 3 vezes nas páginas [3](#), [16](#) e [17](#).
- ROUTRAY, S. K. et al. Statistical model for link lengths in optical transport networks. *Journal of Optical Communications and Networking*, Optical Society of America, v. 5, n. 7, p. 762–773, 2013. Citado na página [44](#).
- SAINI, H.; GARG, A. K. Protection and restoration schemes in optical networks: a comprehensive survey. *International Journal of Microwaves Applications*, v. 2, n. 1, 2013. Citado 2 vezes nas páginas [viii](#) e [16](#).
- SAMBO, N. et al. Next generation sliceable bandwidth variable transponders. *IEEE Communications Magazine*, IEEE, v. 53, n. 2, p. 163–171, 2015. Citado na página [18](#).
- SHEN, G.; GUO, H.; BOSE, S. K. Survivable elastic optical networks: survey and perspective. *Photonic Network Communications*, Springer, v. 31, n. 1, p. 71–87, 2016. Citado na página [18](#).
- SONG, L.; ZHANG, J.; MUKHERJEE, B. Dynamic provisioning with availability guarantee for differentiated services in survivable mesh networks. *IEEE Journal on Selected Areas in Communications*, IEEE, v. 25, n. 3, 2007. Citado na página [6](#).

- SUURBALLE, J. Disjoint paths in a network. *Networks*, Wiley Online Library, v. 4, n. 2, p. 125–145, 1974. Citado 2 vezes nas páginas 4 e 20.
- SUURBALLE, J.; TARJAN, R. E. A quick method for finding shortest pairs of disjoint paths. *Networks*, v. 14, n. 2, p. 325–336, 1984. Citado 3 vezes nas páginas 4, 20 e 25.
- TALEBI, S. et al. Spectrum management techniques for elastic optical networks: A survey. *Optical Switching and Networking*, Elsevier, v. 13, p. 34–48, 2014. Citado 3 vezes nas páginas 13, 14 e 15.
- TESSINARI, R. S.; COLLE, D.; GARCIA, A. S. A defragmentation-ready simulation framework for elastic optical networks. *Journal of Communication and Information Systems*, v. 32, n. 1, 2017. Citado 3 vezes nas páginas xi, 20 e 23.
- TESSINARI, R. S. et al. Zone based spectrum assignment in elastic optical networks: a fairness approach. In: IEEE. *Opto-Electronics and Communications Conference (OECC)*, 2015. [S.l.], 2015. p. 1–3. Citado na página 20.
- TESSINARI, R. S. et al. ElasticO++: An elastic optical network simulation framework for OMNeT++. *Optical Switching and Networking*, Elsevier, v. 22, p. 95–104, 2016. Citado 9 vezes nas páginas viii, 1, 2, 7, 9, 20, 21, 22 e 39.
- THOLEY, T. Linear time algorithms for two disjoint paths problems on directed acyclic graphs. *Theoretical Computer Science*, Elsevier, v. 465, p. 35–48, 2012. Citado na página 4.
- TOMKOS, I. et al. A tutorial on the flexible optical networking paradigm: State of the art, trends, and research challenges. *Proceedings of the IEEE*, IEEE, v. 102, n. 9, p. 1317–1337, 2014. Citado na página 1.
- VARGA, A. *OMNeT++ web page*. 2017. <<https://omnetpp.org/>>. Citado na página 20.
- WANG, Y.; ZENG, Q.; ZHAO, H. Dynamic survivability in WDM mesh networks under dynamic traffic. *Photonic Network Communications*, Springer, v. 6, n. 1, p. 5–24, 2003. Citado 4 vezes nas páginas ix, 6, 42 e 44.
- WHITNEY, H. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, The Johns Hopkins University Press, v. 54, n. 1, p. 150–168, 1932. ISSN 00029327. Citado 2 vezes nas páginas 2 e 25.
- WILLIAMS, R. Finding paths of length k in $O^*(2k)$ time. *Information Processing Letters*, Elsevier, v. 109, n. 6, p. 315–318, 2009. Citado na página 37.
- XU, M. et al. An efficient critical protection scheme for intra-domain routing using link characteristics. *Computer networks*, Elsevier, v. 57, n. 1, p. 117–133, 2013. Citado na página 6.
- YEN, J. Y. An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quarterly of Applied Mathematics*, v. 27, n. 4, p. 526–530, 1970. Citado 3 vezes nas páginas 20, 28 e 31.
- ZHANG, G. et al. A survey on OFDM-based elastic core optical networking. *IEEE Communications Surveys & Tutorials*, IEEE, v. 15, n. 1, p. 65–87, 2013. Citado 2 vezes nas páginas 9 e 12.

ZHANG, J. et al. A new provisioning framework to provide availability-guaranteed service in WDM mesh networks. In: IEEE. *International Conference on Communications, 2003. ICC'03. IEEE*. [S.l.], 2003. v. 2, p. 1484–1488. Citado na página [6](#).

ZHANG, K. et al. Dfdp: a distributed algorithm for finding disjoint paths in wireless sensor networks with correctness guarantee. *International Journal of Distributed Sensor Networks*, SAGE Publications Sage UK: London, England, v. 10, n. 6, p. 258959, 2014. Citado na página [4](#).